Vol. 40 No. 12 Transactions of Information Processing Society of Japan Dec. 1999

Regular Paper

On the Σ_1^b-Definability of Integer Factoring

MITSURU TADA† and HIROKI SHIZUYA††

In this paper, we introduce an approach to cryptology using bounded arithmetic, and we investigate factorization. Factorization supports the security of many kinds of cryptosystems. If it could be efficiently computed, then those systems would not be secure any longer. Since functions that are Σ_1^b-definable in S_2^1 are computable in polynomial time, it is a worthwhile task to try to Σ_1^b-define the factoring function. At present, however, it turns out to be necessary to add some axiom to the theory S_2^1 with respect to primality.

1. Introduction

The RSA system [10] is one of the most popular encryption schemes. Like various other systems (e.g., Rabin, Williams, etc.) its security is supported by the difficulty of factoring a given integer. (See Okamoto [5], etc. for several more examples of such systems.) Since the factoring function used to break the systems above is an operation to compute (p, q) from the input a $(= pq)$, not to compute the perfect factorization $q_1^{e_1} \cdots q_t^{e_t}$ from an arbitrary number a, we call the first operation *the factoring function*. At first, the term *factorization* means the problem of finding the pair of primes (p, q) such that $p \cdot q = n$ for a given number n if such a pair exists. Later we will treat a few types of factoring such as the operation $a \mapsto (p, q)$ for an input a $(= p^2 q)$, which is necessary to break other types of encryption systems, such as ESIGN [2], the public-key cryptosystem introduced in Okamoto and Uchiyama [6]. Any ways of breaking RSA other than factoring the given composite pq are unknown, where the word *breaking* means finding a secret parameter from the public ones. In this paper, we will study the Σ_1^b-*definability* of such types of factoring function.

The systems S_2^i $(i \in \mathbf{N})$ of *bounded arithmetic* are introduced in Buss [1]. It is widely known that the theory of bounded arithmetic is closely related to computational complexity theory. For a typical example, the main theorem of Buss [1] relates S_2^i for $i \geq 1$ to $F\Delta_i^p$, where $F\Delta_i^p$ is the function version of Δ_i^p in the polynomial-time hierarchy. To be precise, if $i \geq 1$, then the class of functions that are Σ_i^b-definable in S_2^i exactly coincides with the class $F\Delta_i^p$. In particular, functions that are Σ_1^b-definable in S_2^1 can be computed in polynomial time, and vice versa. Hence the study of S_2^1 is almost directly related with that of FP, the class of all functions that are polynomial-time-computable.

However, there does not seem to be an S_2^1-proof for Σ_1^b-definability of the factoring function. We will therefore give the theory S_2^1 some axiom with respect to *primality*, which will be named $\wp$, and which is unlikely to be proved in S_2^1. This means that we make S_2^1 more powerful than without $\wp$. Denote the extended theory by $S_2^1 + \wp$. We will Σ_1^b-define the factoring function in $S_2^1 + \wp$. In its proof, the added axiom $\wp$ is quite necessary; it can thus provide some credibility to cryptosystems that rely on the difficulty of factorization for their security.

In Krajíček and Pudlák [4], it is shown that $(1, 1)$-type factoring is *implicitly definable* in $S_2^1 + \Phi$, that is, that if the existence condition of $(1, 1)$-factoring is provable, then its uniqueness condition is also provable by the same theory. The additional axiom Φ denotes one direction of Pratt's theorem, and represents *Pratt's primes* as *primitive primes*. In this paper, however, we will add Pratt's theorem itself to the theory S_2^1 in order to give proofs of both the existence condition and the uniqueness condition for several types of factoring functions. This means that if Pratt's theorem could be proved in the theory S_2^1, then those functions would be computable in polynomial time.

This paper is organized as follows: In Section 2, we will review the fundamental definitions and properties with respect to bounded arithmetic systems. In Section 3, we will describe two definitions for primality, one of which is

† School of Information Science, Japan Advanced Institute of Science and Technology
†† Education Centre for Infomation Processing, Tohoku University

constructed after Pratt's theorem. Sections 4 and 5 will give Σ_1^b-definitions of the $(1,1)$-type and $(2,1)$-type factoring functions, respectively. Section 6 will present various types of factoring function and outline our future work.

2. Preliminaries

In this section, we review the construction of the theory S_2^i, which is the instrument for our purpose. First, we examine the definition of the language of bounded arithmetic and the class of bounded formulae Σ_i^b and Π_i^b ($i \in \mathbf{N}$). After that, we examine the definition of the theory S_2^i and its well-known properties. (See Buss [1] for details.)

The language of bounded arithmetic consists of the constant symbol 0, unary function symbols S, $|*|$ and $\lfloor */2 \rfloor$, binary function symbols $+$, $\cdot$ and $\#$, binary predicate symbols $=$ and $\leq$, logical connectives $\neg$, $\wedge$, $\vee$ and $\supset$, quantifiers $\forall$ and $\exists$, and the auxiliary symbols $($, $)$ and , (commas). Here S means the successor function, that is, $S(x) = x + 1$; $|x|$ means the length of the binary representation of x, that is, $|x| = \lceil \log_2(x+1) \rceil$; and $x\#y$ means $2^{|x||y|}$.

We call quantifiers of the form $(\forall x \leq t)$ or $(\exists x \leq t)$ *bounded quantifiers*. In a special case, if t is the length of some term s, that is, $t = |s|$, then we call them *sharply bounded quantifiers*. Quantifiers of the form $(\forall x)$ or $(\exists x)$ are called unbounded quantifiers. A formula all of whose quantifiers are bounded is called a *bounded formula*.

We define the class Σ_0^b to be the set of all *sharply* bounded formulae. The class Π_0^b is defined to be the same set as Σ_0^b. The classes Σ_i^b and Π_i^b ($i \geq 1$) are constructed analogously to the usual arithmetic hierarchy. Thus, bounded quantifiers and sharply bounded quantifiers play the roles of unbounded quantifiers and bounded quantifiers, respectively.

The inference rules in S_2^i are all of those in LK (due to Gentzen) plus four inference rules with respect to bounded quantifiers, also plus Σ_i^b-PIND, which is of the form:

$$\frac{\Gamma, A(\lfloor a/2 \rfloor) \rightarrow A(a), \Delta}{\Gamma, A(0) \rightarrow A(t), \Delta},$$

where a is a free variable not appearing in the lower sequent, t is an arbitrary term, and A is an arbitrary Σ_i^b-formula. (However, we do not use any induction rules in this paper.) The set of axioms in S_2^i is called BASIC, and consists of 32 open formulae describing fundamental properties of functions and predicates in bounded arithmetic language.

One reason why the theories S_2^i interest us so much is that they have a close relationship to the polynomial-time hierarchy. To be precise, functions that are *Σ_i^b-definable in S_2^i* (see below) belong to $F\Delta_i^p$, the class of functions computable by a polynomial-time machine with access to an oracle from Σ_{i-1}^p, the $(i-1)$-th class of sets in the polynomial-time hierarchy. Furthermore, the converse also holds. By substituting $i = 1$, we can show that functions that are Σ_1^b-definable in S_2^1 are polynomial-time-computable, and vice versa.

Definition 2.1 Let $i \geq 1$. Suppose A is a Σ_i^b-formula and that

$$S_2^i \vdash (\forall \boldsymbol{x})(\exists y \leq t) A(\boldsymbol{x}, y)$$

and

$$S_2^1 \vdash (\forall \boldsymbol{x})(\forall y)(\forall z)[A(\boldsymbol{x}, y) \wedge A(\boldsymbol{x}, z) \supset y = z].$$

Then we say that S_2^1 can Σ_i^b-define the function f such that $(\forall \boldsymbol{x}) A(\boldsymbol{x}, f(\boldsymbol{x}))$ is satisfied; that is, the function f is Σ_i^b-definable in S_2^i. Here $\boldsymbol{x}$ and $(\forall \boldsymbol{x})$ are abbreviations for $(x_1, \ldots, x_n)$ and $(\forall x_1) \cdots (\forall x_n)$, respectively. The notation $S_2^i \vdash A$ means that the formula A can be proved in S_2^i.

We call the first and the second formulae in the statement of this theorem the *existence condition* and the *uniqueness condition*, respectively.

Theorem 2.2 (Buss [1]) A function f is in $F\Delta_i^p$ if and only if the function f is Σ_i^b-definable in S_2^i.

This is a fairly noteworthy property, to be sure, but it is quite inconvenient for us that we have only a small number of languages to discuss S_2^i. This difficulty is resolved by the next definition and theorem.

Definition 2.3 Let f be a new function symbol. We define $\Sigma_i^b(f)$ and $\Pi_i^b(f)$ to be sets of bounded formulae in the language of bounded arithmetic plus the symbol f. These sets of formulae are defined by counting alternations of bounded quantifiers, ignoring the sharply bounded quantifiers, exactly as in the definition of Σ_i^b and Π_i^b.

Theorem 2.4 (Buss [1]) Suppose S_2^i can Σ_i^b-define the function f. Let $S_2^i(f)$ be the theory obtained from S_2^i by adding a new function symbol and adding the defining axiom for f. Then if $i > 0$ and B is a $\Sigma_i^b(f)$-(or a $\Pi_i^b(f)$-) formula, there is a $B^* \in \Sigma_i^b$ (or Π_i^b, respectively) such that $S_2^i(f) \vdash B^* \leftrightarrow B$, where $A \leftrightarrow B$ is

the abbreviation for $(A \supset B) \wedge (B \supset A)$.

With the aid of the above theorems and definitions, when we discuss S_2^1, we can use the symbols of any polynomial-time-computable functions in formulae, and of course, in formulae in induction axioms. Thus the symbols of polynomial-time computable functions are used without further comment.

3. Primality

In this section, we define the predicate of primality in two distinct ways. One is a primitive definition, and the other is Pratt's. Here we regard predicates as total functions from any string to $\{0,1\}$. First we try to define the predicate of primality in S_2^1. The well-known co-NP-definition of primality is as follows:

$$(\forall p \le a)(\forall q \le a)[(a = pq) \supset (p = 1) \vee (q = 1)].$$

Let $\mathsf{Prime}(a)$ be this formula. On the other hand, let $\mathsf{Comp}(a)$ be the predicate for judging whether a is a composite or not, namely:

$$\begin{aligned}\mathsf{Comp}(a) \Leftrightarrow\ & (\exists p \le a)(\exists q \le a)[(a = pq) \\ & \wedge (p \ne 1) \wedge (q \ne 1)] \\ \equiv\ & (\exists p < a)(\exists q < a)[a = pq],\end{aligned}$$

where $x < y$ is the abbreviation for $(x \le y) \wedge \neg(x = y)$. We can easily get $S_2^1 \vdash \mathsf{Comp}(a) \leftrightarrow \mathsf{Prime}(a)$. The predicate of primality is the function that on input of a outputs 1 when a is a prime number and 0 otherwise. We therefore define the formula $P(a,y)$ as follows:

$$(\mathsf{Prime}(a) \wedge y = 1) \vee (\neg\mathsf{Prime}(a) \wedge y = 0).$$

The theory S_2^1 can then prove both the existence and uniqueness conditions. But we cannot say that the predicate of primality is in P, because the defining formula $P(a,y)$ is not known to be in Σ_1^b. The problem is that, since $\mathsf{Prime}(a)$ is in Π_1^b and $\neg\mathsf{Prime}(a)$ is in Σ_1^b, the formula $P(a,y)$ turns out to be in $\Sigma_2^b \cap \Pi_2^b$, which is a larger class than Σ_1^b. What we can say here is that the primality predicate is in Δ_2^p. Hence we need an NP-definition of primality. We therefore use Theorem 10.1 from Papadimitriou [7]:

Theorem 3.1 (Pratt [7),9)]) A number $p > 1$ is prime if and only if there is a number $1 < r < p$ such that $r^{p-1} \equiv 1 (\mathrm{mod}\ p)$, and furthermore $r^{(p-1)/q} \not\equiv 1 (\mathrm{mod}\ p)$ for all prime divisors q of $p-1$.

Pratt showed that the predicate for primality belongs to NP∩co-NP. By this theorem, as in Krajíček and Pudlák [4], we define the formula $C(p,w)$ as follows:

$$\begin{aligned}& w = \langle g, p, q_1, e_1, w_1, \ldots, q_t, e_t, w_t\rangle \\ \wedge\ & g \in (\mathbf{Z}/p\mathbf{Z})^* \wedge g^{p-1} \equiv 1 (\mathrm{mod}\ p) \\ \wedge\ & p - 1 = \textstyle\prod_{i \le t} q_i^{e_i} \\ \wedge\ & (\forall i \le t)[g^{(p-1)/q_i} \not\equiv 1 (\mathrm{mod}\ p)] \\ \wedge\ & (\forall i \le t) C(q_i, w_i).\end{aligned}$$

In the formula above, since t is bounded by the length of p, the quantifiers are essentially sharply bounded. The NP-definition of primality $\mathsf{Pratt}(a)$ is $(\exists w \le t(a))C(a,w)$ for some suitable term $t(a)$. Note that $\mathsf{Pratt}(a)$ is in Σ_1^b, but that $\neg\mathsf{Pratt}(a)$ is not known to be in Σ_1^b, so the Σ_1^b-definition of primality is not yet completed.

We now define Φ to be the formula $(\forall x)[\mathsf{Pratt}(x) \supset \mathsf{Prime}(x)]$. On the other hand, we denote the converse of Φ by Ψ. Thus the equivalence between $\mathsf{Pratt}(a)$ and $\mathsf{Prime}(a)$ can be represented by $\Phi \wedge \Psi$. Pratt showed that Φ and Ψ hold, but it is conjectured that the theory S_2^1 can prove neither Φ nor Ψ. We therefore denote Pratt's theorem by $\wp\ (\Leftrightarrow \Phi \wedge \Psi)$.

4. Factoring (1,1)-Type Composites

Many cryptosystems such as RSA, Rabin and Williams rely upon the difficulty of factoring an integer given at random. Therefore, if a factoring function were included in FP, those systems would no longer be secure. To break the systems, we do not need perfect factoring such that

$$a \mapsto (p_1, \ldots, p_t, e_1, \ldots, e_t),$$

where $a = \prod_{i=1}^t p_i^{e_i}$, each p_i is a prime and $e_i \ge 1$ for each i. Hence we hereafter call the following operation *the* $(1,1)$*-factoring function.*

$$a \mapsto \begin{cases} \langle p, q\rangle, & \text{if } a \text{ is the product of two primes } p \text{ and } q; \\ c_1, & \text{if } a \text{ is the product of } p \text{ and } q \text{ but at least one of them is a composite;} \\ c_2, & \text{if } a \text{ is a prime,} \end{cases}$$

where $\langle p, q\rangle$ is the *sequence number* of a sequence (p,q) defined in Buss [1], and where c_1 and c_2 are distinct constants. We denote this $(1,1)$-factoring function by $\mathsf{Fact}_{(1,1)}$.

Here we try to Σ_1^b-define the function $\mathsf{Fact}_{(1,1)}$ with bounded arithmetic language. Since Φ or Ψ does not seem to be proved in S_2^1 as mentioned in the previous section, we consider the theory $S_2^1 + \wp$, which is obtained by adding Pratt's theorem as an axiom to the theory S_2^1. We can define the defining formula $F_{(1,1)}$ of $\mathsf{Fact}_{(1,1)}$ as follows:

$$\dfrac{\dfrac{\dfrac{((u,v<a)\wedge(a=uv))\wedge \mathsf{Pratt}(u,v)\rightarrow((u,v<a)\wedge(a=uv))\wedge\mathsf{Pratt}(u,v)}{((u,v<a)\wedge(a=uv))\wedge\mathsf{Pratt}(u,v)\rightarrow F'_{(1,1)}(a,y')}}{((u,v<a)\wedge(a=uv))\wedge\mathsf{Pratt}(u,v)\rightarrow F_{(1,1)}(a,y')}}{((u,v<a)\wedge(a=uv))\wedge\mathsf{Pratt}(u,v)\rightarrow(\exists y)F_{(1,1)}(a,y)}$$

Fig. 1 Proof in Theorem 4.2.

$$\begin{aligned}&F_{(1,1)}(a,y)\\ \Leftrightarrow\ &(\exists p,q<a)[(a=pq)\wedge\mathsf{Pratt}(p,q)\\ &\quad\wedge(y=\langle\mathsf{min}(p,q),\mathsf{max}(p,q)\rangle)]\\ &\vee(\exists p,q<a)[(a=pq)\wedge\neg(\mathsf{Prime}(p,q))\\ &\quad\wedge(y=c_1)]\\ &\vee(\mathsf{Pratt}(a)\wedge(y=c_2)).\end{aligned}$$

Here $\mathsf{Pratt}(p,q)$ is the abbreviation for $\mathsf{Pratt}(p)\wedge\mathsf{Pratt}(q)$. In Krajíček and Pudlák[4], it is shown that the uniqueness condition of $\mathsf{Fact}_{(1,1)}$ is proved in the theory $S_2^1+\Phi$.

Theorem 4.1 (Krajíček and Pudlák[4]) The theory $S_2^1+\Phi$ proves the sequent

$$\mathsf{Pratt}(p,q,p',q'),p\le q,p'\le q',\ a=pq,a=p'q'\rightarrow p=p'\wedge q=q'.$$

By means of the next theorem, we can show that the $(1,1)$-factoring function $\mathsf{Fact}_{(1,1)}$ is Σ_1^b-definable in $S_2^1+\wp$.

Theorem 4.2 The existence condition of $\mathsf{Fact}_{(1,1)}$ can be proved in $S_2^1+\Psi$.

Proof.

First, we prove the following sequent in S_2^1:

$$\neg\mathsf{Prime}(a)\rightarrow(\exists y)F_{(1,1)}(a,y).$$

This proof is constructed as **Fig. 1**. In the proof, a double-underline means that several inference rules are applied. Here y' is the term $\langle\mathsf{min}(u,v),\mathsf{max}(u,v)\rangle$, $(u,v<a)$ are the abbreviations for $(u<a)\wedge(v<a)$, and $F_2'(a,b)$ is the following formula:

$$\begin{aligned}&(\exists p<a)(\exists q<a)[(a=pq)\wedge\mathsf{Pratt}(p,q)\\ &\qquad\wedge(b=\langle\mathsf{min}(p,q),\mathsf{max}(p,q)\rangle)].\end{aligned}$$

Note that we have used the fact $S_2^1\vdash(\exists x\le t)A(x)\leftrightarrow(\exists x)[(x\le t)\wedge A(x)]$. By almost the same proof, we can demonstrate that the following sequent is proved in S_2^1:

$$\begin{aligned}&((u,v<a)\wedge(a=uv))\wedge\neg\mathsf{Pratt}(u,v)\\ &\qquad\rightarrow(\exists y)F_{(1,1)}(a,y).\end{aligned}$$

Hence S_2^1 can prove $\neg\mathsf{Prime}(a)\rightarrow(\exists y)F_{(1.1)}(a,y)$, since $\mathsf{Comp}(a)\equiv\neg\mathsf{Prime}(a)$.

Next we prove that the sequent $\mathsf{Pratt}(a)\rightarrow(\exists y)F_{(1,1)}(a,y)$ can be proved in S_2^1. This proof is quite simple:

$$\dfrac{\dfrac{\dfrac{\mathsf{Pratt}(a)\rightarrow\mathsf{Pratt}(a)}{\mathsf{Pratt}(a)\rightarrow\mathsf{Pratt}(a)\wedge(c_2=c_2)}}{\mathsf{Pratt}(a)\rightarrow F_{(1,1)}(a,c_2)}}{\mathsf{Pratt}(a)\rightarrow(\exists y)F_{(1,1)}(a,y)}$$

Hence S_2^1 can prove the sequent $\neg\mathsf{Prime}(a)\vee\mathsf{Pratt}(a)\rightarrow(\exists y)F_{(1,1)}(a,y)$. Since Ψ asserts $\neg\mathsf{Prime}(a)\vee\mathsf{Pratt}(a)$, the theory $S_2^1+\Psi$ can prove $(\exists y)F_{(1,1)}(a,y)$. From Parikh's theorem[1], for some term $t(a)$, $S_2^1+\Psi$ can prove $(\exists y\le t(a))F_{(1,1)}(a,y)$. ■

We now have the uniqueness condition (Theorem 4.1) and the existence condition (Theorem 4.2), and can therefore obtain the next corollary:

Corollary 4.3 The $(1,1)$-factoring function $\mathsf{Fact}_{(1,1)}$ is Σ_1^b-definable in $S_2^1+\wp$.

5. Factoring (2,1)-Type Composites

In the previous section, we discussed factoring that is the mapping $a\mapsto(p,q)$, where a is the product of two primes p and q. Since RSA, Rabin, etc. have $n\ (=pq)$ as a public key and (p,q) as secret keys, it is quite significant to investigate the function $\mathsf{Fact}_{(1,1)}$.

In this section, we consider the case of an attempt to break ESIGN, which is introduced in Fujioka, et al.[2]. Here we say that ESIGN is broken if the secret keys p and q are obtained from the public key $n\ (=p^2q)$. Thus we need to investigate the function $\mathsf{Fact}_{(2,1)}$ which is the mapping $a\mapsto\langle p,q\rangle$, where for two primes p and q, a is the product of two p's and q, that is, $a=p^2q$. As well as $\mathsf{Fact}_{(1,1)}$, we try to Σ_1^b-define the function $\mathsf{Fact}_{(2,1)}$ in the theory $S_2^1+\wp$.

The defining formula of $\mathsf{Fact}_{(2,1)}$, $F_{(2,1)}(a,y)$ is given by:

$$\begin{aligned}&F_{(2,1)}(a,y)\\ \Leftrightarrow\ &(\exists p,q<a)[(a=pq)\wedge\mathsf{R}(p)\wedge\mathsf{Pratt}(\sqrt{p},q)\\ &\qquad\wedge(y=\langle\mathsf{min}(\sqrt{p},q),\mathsf{max}(\sqrt{p},q)\rangle)]\\ &\vee(\exists p,q<a)[(a=pq)\wedge\mathsf{R}(p)\\ &\qquad\wedge\neg\mathsf{Prime}(\sqrt{p},q)\wedge(y=c_1)]\\ &\vee(\exists p,q<a)[(a=pq)\wedge\neg\mathsf{R}(p)\wedge(y=c_2)]\\ &\vee(\mathsf{Pratt}(a)\wedge(y=c_3)),\end{aligned}$$

where c_1, c_2, and c_3 are different constants, $\mathsf{R}(a)$ is the predicate for judging whether a is a square of some polynomial-time-computable number, that is Σ_1^b-definable in S_2^1, and $\sqrt{*}$ is the polynomial-time-computable function for obtaining the integer part of the square root of

$$\dfrac{\dfrac{\dfrac{\dfrac{(u,v<a)\wedge(a=uv)\wedge \mathsf{R}(u)\wedge \mathsf{Pratt}(\sqrt{u},v)\rightarrow (u,v<a)\wedge(a=uv)\wedge \mathsf{R}(u)\wedge \mathsf{Pratt}(\sqrt{u},v)}{A(a,u,v)\rightarrow (u,v<a)\wedge(a=uv)\wedge \mathsf{R}(u)\wedge \mathsf{Pratt}(\sqrt{u},v)\wedge(\mathsf{M}(\sqrt{u},v)=\mathsf{M}(\sqrt{u},v))}}{A(a,u,v)\rightarrow (\exists p,q<a)[(a=pq)\wedge \mathsf{R}(p)\wedge \mathsf{Pratt}(\sqrt{p},q)\wedge(\mathsf{M}(\sqrt{u},v)=\mathsf{M}(\sqrt{p},q))]}}{A(a,u,v)\rightarrow F_{(2,1)}(a,\mathsf{M}(\sqrt{u},v))}}{A(a,u,v)\rightarrow (\exists y)F_{(2,1)}(a,y)}$$

Fig. 2 Proof in Theorem 5.3.

the input. To be precise, $\mathsf{R}(a) \Leftrightarrow a = (\sqrt{a})^2$. $F_{(2,1)}$ is thus a Σ^b_1-formula.

To Σ^b_1-define the function $F_{(2,1)}$, we have to show that $S^1_2 + \wp$ can prove both the existence condition and the uniqueness condition. The uniqueness condition is obtained by improving Theorem 4.1 slightly.

Theorem 5.1 The theory $S^1_2+\Phi$ proves the following sequent:

$$\mathsf{Pratt}(p,q,r,p',q',r'), p\le q\le r, p'\le q'\le r',$$
$$a = pqr, a = p'q'r'$$
$$\rightarrow (p=p')\wedge(q=q')\wedge(r=r'),$$

where $\mathsf{Pratt}(p,q,r,p',q',r')$ is the abbreviation for $\mathsf{Pratt}(p)\wedge\mathsf{Pratt}(q)\wedge\mathsf{Pratt}(r)\wedge\mathsf{Pratt}(p')\wedge\mathsf{Pratt}(q')\wedge\mathsf{Pratt}(r')$.

Proof.

First, we show that S^1_2 proves the sequent $\mathsf{Pratt}(p)\wedge(p|abc)\rightarrow(p|a)\vee(p|b)\vee(p|c)$. Assume that $\mathsf{Pratt}(p)$ and $p|abc$, but that p does not divide a or b. Let $\gcd(x,y)$ be the greatest common divisor of x and y, which is efficiently obtained. Then $\gcd(a,p)$ must be equal to 1, and hence we get

$$pu + av = 1,$$

for some u, v, and so also

$$pubc + avbc = bc.$$

Since $p|abc$, p divides the left-hand side and hence $p|bc$. Furthermore, $\gcd(b,p)$ must also be equal to 1. Therefore we can get $p|c$ in a similar manner.

Let $pqr = p'q'r'$. Then $p|p'q'r'$ and from the statement before, $p|p'$, $p|q'$, or $p|r'$. The same holds for q and r. Therefore, from the assumptions $p\le q\le r$ and $p'\le q'\le r'$, we get $p=p'$, $q=q'$ and $r=r'$. ■

Similarly, we can show the following corollary:

Corollary 5.2 Let c be a constant. Then the theory $S^1_2+\Phi$ proves the sequent:

$$\mathsf{Pratt}(p_1,\ldots,p_c,p'_1,\ldots,p'_c), p_1\le\cdots\le p_c,$$
$$p'_1\le\cdots\le p'_c, a=p_1\cdots p_c, a=p'_1\cdots p'_c$$
$$\rightarrow (p_1=p'_1)\wedge\cdots\wedge(p_c=p'_c).$$

Theorem 5.3 The function $\mathsf{Fact}_{(2,1)}$ is Σ^b_1-definable in $S^1_2+\wp$.

Proof.

The uniqueness condition is proved by the previous theorem. We have only to show the existence condition of $\mathsf{Fact}_{(2,1)}$, whose proof can be constructed in almost the same way as in the case of $\mathsf{Fact}_{(1,1)}$ as **Fig. 2**. In that proof tree, $A(a,s,t)$ and $\mathsf{M}(s,t)$ denote the formula $(s,t<a)\wedge(a=st)\wedge\mathsf{R}(s)\wedge\mathsf{Pratt}(\sqrt{s},t)$ and $\langle\mathsf{min}(s,t),\mathsf{max}(s,t)\rangle$, respectively.

Similarly, we can show that S^1_2 can prove the sequents:

$$(u,v<a)\wedge(a=uv)\wedge\mathsf{R}(u)$$
$$\wedge\neg\mathsf{Prime}(\sqrt{u},v)\rightarrow(\exists y)F_{(2,1)}(a,y),$$
$$(u,v<a)\wedge(a=uv)\wedge\neg\mathsf{R}(u)$$
$$\rightarrow(\exists y)F_{(2,1)}(a,y),$$

and

$$\mathsf{Pratt}(a)\rightarrow(\exists y)F_{(2,1)}(a,y).$$

We can show that S^1_2 proves $\neg\mathsf{Prime}(a)\vee\mathsf{Pratt}(a)\rightarrow(\exists y)F_{(2,1)}(a,y)$. The added axiom Ψ asserts $\neg\mathsf{Prime}(a)\vee\mathsf{Pratt}(a)$, and hence we can conclude that $S^1_2+\Psi\vdash(\exists y)F_{(2,1)}(a,y)$. From Parikh's theorem, for some term $t(a)$, $S^1_2+\Psi\vdash(\exists y\le t(a))F_{(2,1)}(a,y)$. ■

We can easily see that the functions $\mathsf{Fact}_{(1,1)}$ and $\mathsf{Fact}_{(2,1)}$ can also act as the predicates for primality, and that neither Φ nor Ψ is necessary to Σ^b_2-define $\mathsf{Fact}_{(1,1)}$ or $\mathsf{Fact}_{(2,1)}$ in S^2_2. In that case, the only result we can get is that $\mathsf{Fact}_{(1,1)}$ and $\mathsf{Fact}_{(2,1)}$ are in $F\Delta^p_2$. Actually, they are in $\mathrm{FP}^{\Sigma^p_1}[wit, O(\log n)]$, where $\mathrm{FP}^{\Sigma^p_1}[wit, O(\log n)]$ is the class of multivalued functions f computable by a polynomial-time witness-oracle Turing machine such that (1) on an input of length n the machine makes at most $O(\log n)$ oracle queries, (2) the witness-oracle has the form $(\exists y\le t(a))R(a,y)$ with $R\in\Delta^b_1$, and (3) on an input x the machine outputs some y such that $f(x)=y$. Furthermore, from the uniqueness of factorization, $\mathsf{Fact}_{(1,1)}$ and $\mathsf{Fact}_{(2,1)}$ are NPSV functions (single-valued functions computed by nondeterministic polynomial time-bounded transducers). Therefore we can conclude that they are in the intersection of NPSV and $\mathrm{FP}^{\Sigma^b_1}[wit, O(\log n)]$.

We use Ψ to make the formulae $F_{(1,1)}$ and $F_{(2,1)}$ belong to Σ^b_1, and we use Φ to prove

the uniqueness of $F_{(1,1)}$ and $F_{(2,1)}$. If $\wp$ could be proved by S_2^1 itself, then by Buss' theorem [1], there would exist polynomial-time algorithms for computing the functions $\mathsf{Fact}_{(1,1)}$ and $\mathsf{Fact}_{(2,1)}$. We can see the ordinary proof of $\wp$ in Ref. 7), whereas it must be quite difficult to give its proof in the theory S_2^1. Recall that the function symbols we can use in proofs in S_2^1 are restricted to those in FP, but that some symbols of functions not known to be in FP appear in the Pratt's proof of $\mathsf{Pratt}(a) \leftrightarrow \mathsf{Prime}(a)$. Thus many cryptosystems based on the difficulty of integer factoring are still kept secure, and if Pratt's theorem could be proved in S_2^1, then the security of those systems would collapse. This provides some credibility to the security of many cryptosystems in the past, present, and future.

6. Generalization of Factoring Function

We have seen that both of the factoring functions $\mathsf{Fact}_{(1,1)}$ and $\mathsf{Fact}_{(2,1)}$ are Σ_1^b-definable in the theory $S_2^1 + \wp$. In other words, if Pratt's theorem could be proved in S_2^1, then these functions would be computable in polynomial-time. This implies that many kinds of cryptosystems whose security relies upon the difficulty of computing these functions could be broken. Here the word 'break' means finding the secret keys from the public key(s). In this section, we extend these functions so that they can be applied to many kinds of composites.

6.1 Factoring (k_1, k_2)-Type Composites

First, we consider factoring a of the form $p_1^{k_1} p_2^{k_2}$, where p_1 and p_2 are primes and k_1 and k_2 are constants. We call this function $\mathsf{Fact}_{(k_1,k_2)}$, and let $F_{(k_1,k_2)}$ be its defining axiom. By improving $F_{(2,1)}$, we can easily define $F_{(k_1,k_2)}$.

$$
\begin{aligned}
&F_{(k_1,k_2)} \Leftrightarrow \\
&(\exists p, q < a)[(a = pq) \wedge (\mathsf{R}_{k_1}(p) \wedge \mathsf{R}_{k_2}(q)) \\
&\wedge \mathsf{Pratt}(\sqrt[k_1]{p},\ \sqrt[k_2]{q}) \\
&\wedge (y = \langle \mathsf{min}(\sqrt[k_1]{p},\ \sqrt[k_2]{q}), \mathsf{max}(\sqrt[k_1]{p},\ \sqrt[k_2]{q})\rangle)] \\
&\vee (\exists p, q < a)[(a = pq) \wedge (\mathsf{R}_{k_1}(p) \wedge \mathsf{R}_{k_2}(q)) \\
&\wedge \neg \mathsf{Prime}(\sqrt[k_1]{p},\ \sqrt[k_2]{q}) \wedge (y = c_1)] \\
&\vee (\exists p, q < a)[(a = pq) \wedge \neg(\mathsf{R}_{k_1}(p) \wedge \mathsf{R}_{k_2}(q)) \\
&\wedge (y = c_2)] \\
&\vee (\mathsf{Pratt}(a) \wedge (y = c_3)),
\end{aligned}
$$

where c_1, c_2, and c_3 are distinctive constants. The predicate $\mathsf{R}_k(a)$ is defined as

$$\mathsf{R}_k(a) \Leftrightarrow a = (\sqrt[k]{a})^k.$$

Here $\sqrt[k]{a}$ means the integer part of the k-th root of a. Since $\sqrt[k]{*}$ is an FP function, the predicate R_k is in P. Then $F_{(k_1,k_2)}$ is a Σ_1^b-formula. The existence condition is proved in almost the same way as Theorem 5.3. The uniqueness condition is obtained from Corollary 5.2 by $c = k_1 + k_2$. Therefore the function $\mathsf{Fact}_{(k_1,k_2)}$ is Σ_1^b-definable in $S_2^1 + \wp$.

6.2 Factoring $(1, 1, 1)$-Type Composites

Here we consider factoring a of the form pqr, where p, q, and r are primes. This is the general case of factoring $(2, 1)$-type composites. We call this function Fact_3 instead of $\mathsf{Fact}_{(1,1,1)}$. (Thus we may say Fact_2 instead of $\mathsf{Fact}_{(1,1)}$.) The defining formula of Fact_3, F_3 is defined as follows:

$$
\begin{aligned}
&F_3(a, y) \Leftrightarrow \\
&(\exists p, q < a)(\exists p_1, p_2 < p)[(a = pq) \wedge (p = p_1 p_2) \\
&\wedge \mathsf{Pratt}(p_1, p_2) \wedge \mathsf{Pratt}(q) \\
&\wedge (y = \langle \mathsf{min}(p_1, p_2, q), \mathsf{mid}(p_1, p_2, q), \\
&\qquad\qquad \mathsf{max}(p_1, p_2, q)\rangle)] \\
&\vee (\exists p, q < a)(\exists p_1, p_2 < p)[(a = pq) \wedge (p = p_1 p_2) \\
&\wedge \neg \mathsf{Prime}(p_1, p_2) \wedge \mathsf{Pratt}(q) \wedge (y = c_1)] \\
&\vee (\exists p, q < a)(\exists q_1, q_2 < q)[(a = pq) \wedge (q = q_1 q_2) \\
&\wedge \mathsf{Pratt}(q_1, q_2) \wedge \mathsf{Pratt}(p) \\
&\wedge (y = \langle \mathsf{min}(p, q_1, q_2), \mathsf{mid}(p, q_1, q_2), \\
&\qquad\qquad \mathsf{max}(p, q_1, q_2)\rangle)] \\
&\vee (\exists p, q < a)(\exists q_1, q_2 < q)[(a = pq) \wedge (q = q_1 q_2) \\
&\wedge \neg \mathsf{Prime}(q_1, q_2) \wedge \mathsf{Pratt}(p) \wedge (y = c_1)] \\
&\vee (\exists p, q < a)[(a = pq) \wedge \mathsf{Comp}(p, q) \wedge (y = c_1)] \\
&\vee (\exists p, q < a)[(a = pq) \wedge \mathsf{Pratt}(p, q) \\
&\wedge (y = \langle \mathsf{min}(p, q), \mathsf{max}(p, q)\rangle)] \\
&\vee (\mathsf{Pratt}(a) \wedge (y = c_2)),
\end{aligned}
$$

where $\mathsf{mid}(a, b, c)$ is the second maximum value of (a, b, c); that is, for example, if $a \le c \le b$, then $\mathsf{mid}(a, b, c) = c$. Since mid is an FP function, F_3 is in Σ_1^b. The formula $(\exists y \le t(a))F_3(a, y)$ is proved as well as $F_{(1,1)}$. The uniqueness condition is obtained by Theorem 5.1. Thus the function Fact_3 is Σ_1^b-definable in $S_2^1 + \wp$. As we can easily guess, the performance of Fact_3 includes that of $\mathsf{Fact}_{(1,1)}$ and that of $\mathsf{Fact}_{(2,1)}$.

For a constant c, we can also Σ_1^b-define the function Fact_c, which is the mapping $a \mapsto (p_1, \ldots, p_c)$, where $a = p_1 \cdots p_c$ and $p_1, \ldots, p_c$ are primes. If we let F_c be the defining formula, the number of lines of F_c increases exponentially with respect to c, but since c is a constant, we can define it.

6.3 Further Generalization and Open Questions

We have seen that factoring functions of the forms:

$a \mapsto \langle p, q \rangle$,
where $a = pq$, and p and q are primes,
$a \mapsto \langle p, q \rangle$,
where $a = p^2 q$, and p and q are primes,
$a \mapsto \langle p_1, \ldots, p_c \rangle$,
where $a = p_1 \cdots p_c$, each p_i is prime,
and where c is a constant,

are all Σ_1^b-definable in $S_2^1 + \wp$. At present, it is not clear whether Pratt's theorem can be proved in S_2^1. Here we try to define functions for factoring other types of composites.

First we consider the (e_1, e_2)-factoring function $\mathsf{Fact}_{(e_1,e_2)}$, where e_1 and e_2 are not always constants. Before we define it, we need the following algorithm.

Algorithm 6.1 This algorithm computes the predicate $\mathsf{Ppow}(e, a)$ and the function $\mathsf{fpow}(e, a)$, which recognizes if $a = p^e$ for some p, and finds p such that $a = p^e$, respectively.

Input: e, (where $0 < e \leq |a|$), a.
Step 1: $m \leftarrow a$; $n \leftarrow 1$.
Step 2: $n \leftarrow n + 1$; $b \leftarrow n^m (\mathrm{mod}\ m)$; $p \leftarrow \gcd(b - n, m)$.
Step 3: If $(p \neq 1) \wedge (p^e \neq a)$, then $m \leftarrow p$ and go back to Step 2. If $(p \neq 1) \wedge (p^e = a)$, then $d \leftarrow 1$. If $p = 1$, then $d \leftarrow 0$.
Output: d, p.

Note that modular powering $(a, b, c) \mapsto a^b (\mathrm{mod}\ c)$ is polynomial-time-computable, and that p^e in Step 3 is also, because e is bounded by a length $|a|$. Thus $\mathsf{Ppow}(e, a)$ is 1, if there exists some p such that $p^e = a$, and then $\mathsf{fpow}(e, a) = p$. Otherwise, $\mathsf{Ppow}(e, a)$ is 0, in which case $\mathsf{fpow}(e, a)$ can be defined arbitrarily. By using these P-predicate and FP-function, we can define $F_{(e_1,e_2)}$, which is the defining formula of the function $\mathsf{Fact}_{(e_1,e_2)}$, as follows:

$$
\begin{aligned}
&F_{(e_1,e_2)}(a, y) \Leftrightarrow \\
&(\exists p, q < a)(\exists e_1 \leq |p|)(\exists e_2 \leq |q|)[(a = pq) \\
&\wedge(\mathsf{Ppow}(e_1, p) \wedge \mathsf{Ppow}(e_2, q) \\
&\wedge\mathsf{Prime}(\mathsf{fpow}(e_1, p), \mathsf{fpow}(e_2, q))) \\
&\wedge(y = \langle \mathsf{fpow}(e_1, p), \mathsf{fpow}(e_2, q), e_1, e_2 \rangle)] \\
&\vee(\exists p, q < a)(\forall e_1 \leq |p|)(\forall e_2 \leq |q|)[(a = pq) \\
&\wedge\neg(\mathsf{Ppow}(e_1, p) \wedge \mathsf{Ppow}(e_2, q) \\
&\wedge\mathsf{Prime}(\mathsf{fpow}(e_1, p), \mathsf{fpow}(e_2, q))) \wedge (y = c_1)] \\
&\vee(\mathsf{Prime}(a) \wedge (y = c_2)).
\end{aligned}
$$

Of course, we can make this formula belong to Σ_1^b by using Ψ. Similarly, we can define the defining formula $F_{(e_1,\ldots,e_c)}$ of the function $\mathsf{Fact}_{(e_1,\ldots,e_c)}$ such that $\mathsf{Fact}_{(e_1,\ldots,e_c)}(a) = \langle p_1, \ldots, p_c, e_1, \ldots, e_c \rangle$, where $a = p_1^{e_1} \cdots p_c^{e_c}$, p_i is prime and $e_i > 0$ for each i, and c is a constant. Then the existence condition of $\mathsf{Fact}_{(e_1,\ldots,e_c)}$ is proved in $S_2^1 + \Psi$, even if we translate the formula above into a Σ_1^b-formula by using Pratt. Note that the uniqueness condition is not yet completed. Unlike the previous factoring functions, this type factoring does not declare the number of prime factors of a given input. Remember that for each function we have given, the number of prime factors of an input is always fixed. But for the function $\mathsf{Fact}_{(e_1,\ldots,e_c)}$, it is not. This makes the uniqueness condition a little troublesome to prove in the theory $S_2^1 + \wp$. We therefore leave it as an open question. What we can demonstrate in this way is that the above functions are in $\mathrm{NPSV} \cap \mathrm{FP}^{\Sigma_1^p}[wit, O(\log n)]$. If the uniqueness condition is proved in $S_2^1 + \wp$, then it follows that these functions are Σ_1^b-definable in $S_2^1 + \wp$. Furthermore, if Pratt's theorem could be proved in S_2^1, then by Buss' theorem we could get a polynomial-time algorithm for computing these functions, in which case the security of many cryptosystems would collapse. Although it remains an open question whether Pratt's theorem can be proved in S_2^1, it is conjectured that it cannot.

Acknowledgments The authors would like to thank Eiji Okamoto and Mike Burmester for their invaluable and helpful suggestions and encouragements.

References

1) Buss, S.R.: *Bounded Arithmetic*, Bibliopolis, Napoli (1986).
2) Fujioka, A., Okamoto, T. and Miyaguchi, S.: ESIGN: An efficient digital signature implementation for smart cards, *Advances in Cryptology (Proc. Eurocrypt '91)*, Lecture Notes in Computer Science, Vol.547, pp.446–457, Springer-Verlag (1991).
3) Krajíček, J.: *Bounded Arithmetic, Propositional Logic and Complexity Theory*, Cambridge University Press (1995).
4) Krajíček, J. and Pudlák, P.: Some consequences of cryptographical conjectures for S_2^1 and EF, Leivant, D. (Ed.), *Logic and Computational Complexity*, Lecture Notes in Computer Science, Vol.960, pp.210–220, Springer-Verlag (1994).
5) Okamoto, E.: *An Introduction to the Theory of Cryptography*, Kyouritsu Shuppan (in Japanese) (1996).
6) Okamoto, T. and Uchiyama, S.: A new public-

key cryptosystem as secure as factoring, Advances in Cryptology (*Proc. Eurocrypt '98*), Lecture Notes in Computer Science, Vol.1403, pp.308–318, Springer-Verlag (1998).
7) Papadimitriou, C.H.: *Computational Complexity*, Addison-Wesley (1994).
8) Parikh, R.J.: Existence and feasibility in arithmetic, *Journal of Symbolic Logic*, Vol.36, pp.494–508 (1971).
9) Pratt, V.R.: Every prime has a succinct certificate, *SIAM J. Comput.*, Vol.4, No.3, pp.214–220 (1975).
10) Rivest, R., Shamir, A. and Adleman, L.: A method for obtaining digital signatures and a public-key cryptosystem, *Comm. ACM*, Vol.21, pp.120–126 (1978).

(Received April 15, 1999)
(Accepted September 2, 1999)

Mitsuru Tada was born in Kyoto, Japan, on September 18, 1969. He received the B.S., M.I.S., and Dr.I.S. degrees from Tohoku University, Japan, in 1992, 1995, and 1998, respectively. He joined Japan Advanced Institute of Science and Technology (JAIST) in 1998, and is currently Associate of School of Information Science, JAIST. His interests are on Mathematical logic, Cryptology and Computational Complexity Theory. Dr. Tada is a member of IPSJ.

Hiroki Shizuya was born in Sendai, Japan, on September 14, 1957. He received the B.E., M.E., and Dr.Eng. degrees from Tohoku University, Japan, in 1981, 1984, and 1987, respectively. He joined Tohoku University in 1987, and is currently Professor of both Education Center for Information Processing and Department of Computer and Mathematical Sciences, Graduate School of Information Sciences. His interests are on Cryptology and Computational Complexity Theory. Dr. Shizuya is a member of ACM, IEEE and IEICE.