整数质因子分解算法新进展与传统密码学面临的挑战

董 青 吴 楠
(南京大学计算机软件新技术国家重点实验室 南京 210093)

摘 要 大整数的质因子分解研究是现代数论领域的一个重要课题,其中涉及很多开问题。随着信息时代的来临,大整数质因子分解的复杂性更成为现代密码学的重要理论基础。著名的RSA公钥密码系统的安全性即建立在解决此问题的困难性之上。本文系统地综述了现代理论计算机科学研究中提出的几种解决该问题的新算法,并介绍了量子计算机高效解决此问题的原理和实现方式。最后,本文讨论了在未来量子计算时代传统密码学所面临的挑战并展望了量子密码学的前景。

关键词 整数因子分解,算法,复杂度,数据安全,量子计算机,量子算法,量子密码学

Recent Progress of Integer Factorization Algorithms and Challenges Faced by the Traditional Cryptology

DONG Qing WU Nan
(State Key Laboratory of Novel Software Technology, Nanjing University, Nanjing 210093, China)

Abstract Integer factorization is one of the most important fields in modern number theory, and in this field there are still many open problems. When facing the information era, the complexity of the integer factorization plays a key role and is an important theoretical basis of modern cryptology. The security of the famous RSA public-key cryptosystem is typically based on the difficulty of this problem. This paper surveys several traditional algorithms of number theory for solving the integer factorization problem which are published recently, and gives a brief introduction of quantum computation and the effective quantum algorithm for factorization. Finally, we discuss challenges faced by traditional cryptology and prospects of the quantum cryptology in the new era of quantum computation.

Keywords Integer factorization, Algorithm, Complexity, Data security, Quantum computer, Quantum algorithm, Quantum cryptology

1 引言

整数数论的基本原理指出,任何大于1的正整数均可表示为有限个素数乘积形式。大整数的质因子分解问题是一个典型的单向问题(One-way Problem)[1],其重要性不言而喻。但该问题目前被认为是困难的(Hard Problem)[2]。几百年以来,许多数学家致力于寻找更快的整数因子分解算法。

20世纪晚期,互联网技术的高度普及和电子商务的成功带来了计算机科学的又一次革命,这就是以保护信息安全为主要目的现代密码学的诞生。现代密码学中最成功、应用最为广泛的密码体系之一便是RSA公钥密码系统。特别是21世纪初,随着MD5,SHA-1等一系列基于Hash函数的密码体系的冲突函数算法被我国科学家相继找到[3,4],Hash函数这个被认为是"万灵药"的理论基础正面临着巨大的挑战。RSA算法被认为是现代密码学所能坚守的"最后的城池"。而RSA公钥密码体系被破解的困难性就典型地取决于大整数质因子分解的难度。同时,对RSA密码系统进行攻击的常用手段之一,就是分解其RSA模数。

迄今为止,经典的大整数分解算法已有多种,其中最有效的是数域筛法,它的时间复杂度随待分解整数长度的增加呈指数级增加,这种指数复杂度的算法在经典计算模型(确定性图灵机模型)下被认为是低效的。1994年,Peter Shor提出了整数因子分解的量子算法(简称Shor算法)[18],其渐进时间复杂度是待分解整数长度的多项式,这意味着如果在量子计算机上执行Shor算法,将可以在很短时间内分解很大的整数,这将直接导致RSA公钥密码系统安全性的崩溃。

本文对现有的传统整数因子分解算法及其时间复杂度进行了系统的分析和综述,并简述Shor算法的原理和实现方式。

2 传统的整数分解算法及其复杂度分析

2.1 蒙特卡罗方法

1975年,J. M. Pollard提出了一种通过伪随机函数的周期性进行因子分解的蒙特卡罗算法,称作rho算法[5],对于寻找合数的小因子非常有效。

Rho算法用一个函数 $f: Z/nZ \to Z/nZ$ 作为伪随机函数,x_0 为种子,生成随机数序列 $x_0, x_1, x_2, \cdots$,通过计算 $\gcd(|x_i - x_j|, n)$ 来寻找 n 的因子。由于 Z/nZ 是有限集,那么该序列必然是循环的,即存在 $i \neq j$,使得 $x_i = x_j$,用Floyd循环查找算法来判断循环的出现,于是可构造出如下算法:

1. $a \leftarrow x_0$; $b \leftarrow x_0$; $d \leftarrow 1$;
2. 当 $d=1$ 时重复如下操作:
2.1 $x \leftarrow f(x)$;
2.2 $y \leftarrow f(f(y))$;
2.3 $d \leftarrow \gcd(|a-b|, n)$;
3. 若 $d=n$,则算法失败,终止;否则算法成功,返回 d。

其中,常用的随机函数为 $f(x) = x^2 + c \bmod n$, $c \neq 0, -2$。若采用上述形式的随机函数,则Rho算法找到 n 的因子平均要经过 $O(\sqrt{p})$ 次模乘,那么算法的时间复杂度为 $O(\sqrt{p}$

$(\log N)^2)$。

1980年，R. P. Brent提出了rho算法的一种改进[6]，采用了一种新的循环查找算法 B_Q，代替了Floyd算法，而使分解算法的效率提高了大约24%。

2.2 $p-1$算法，$p+1$算法，椭圆曲线算法

Pollard $p-1$算法利用费马小定理进行因子分解，可以找到一个合数满足如下条件的因子 p：$p-1$ 关于一个相对较小的界 B 是光滑的(整数 m 的所有素因子不大于 B 时，称 m 关于界 B 光滑)。该算法由J. M. Pollard于1974年提出[7]。

对于光滑界 B，Q 是所有不大于 B 的素数的不大于 n 的方幂的最小公倍数。若素数 $q\leqslant B$，$q^l\leqslant n$，则 $l\ln q\leqslant\ln n$，并且 $l\leqslant\lfloor \ln n/\ln q\rfloor$，可知 $Q=\prod_{q\leqslant B}q^{\lfloor \ln n/\ln q\rfloor}$。若 p 是 n 的素因子，且关于界 B 光滑，则 $p-1\mid Q$，因此由费马小定理可知，对任意满足 $\gcd(a,p)=1$ 的整数 a，有 $a^Q\equiv 1\pmod p$。令 $d=\gcd(a^Q-1,n)$，则 $p\mid d$。根据上述思想，可构造出如下算法：

1. 选择光滑界 B；
2. 选取整数 $a\in[2,n-1]$，$d\leftarrow\gcd(a,n)$，若 $d>1$ 则返回 d。
3. 对每个质数 $q\leqslant B$，执行如下操作：
   3.1 $l\leftarrow\lfloor \ln n/\ln q\rfloor$；
   3.2 $a\leftarrow a^{q^l}\bmod n$；
4. $d\leftarrow\gcd(a-1,n)$；
5. 若 $d=1$ 或 $d=n$，则算法失败，终止；否则算法成功，返回 d。

上述算法的时间复杂度为 $O(B\log B\log^2 n)$。实际应用中，光滑界 B 的选择取决于使用者愿意花费在该算法上的时间，一般在 10^5 到 10^6 之间。

1982年，H. C. Williams提出了与Pollard $p-1$ 算法思想类似的Williams $p+1$ 算法[8]，可以找到合数的因子 p，只要 $p+1$ 关于某个界是光滑的。与Pollard $p-1$ 算法中计算 a^Q-1 不同，Williams $p+1$ 算法使用了Lucas序列：

$V_0=2,V_1=a,V_i=aV_{i-1}-V_{i-2}$

令 $D=a^2-4$，对奇素数 p，若 $p-(D/p)\mid M$，则 $p\mid\gcd(V_M-2,n)$，其中 (D/p) 为勒让德符号。由此，可由 $\gcd(V_M-2,n)$ 求得 n 的因子。若 $(D/p)=+1$，则该算法退化为Pollard $p-1$ 算法的慢速版本，而我们希望 $(D/p)=-1$，但因为预先不知道 p 的值，所以可能要尝试多个 a 的值，从而得到 n 的因子。

另一种重要的因子分解方法——椭圆曲线分解法(ECM)也可以看作Pollard p−1算法的改进，它由H. W. Lenstra于1987年提出[9]。Z_p 上的乘法群 Z_p^* 是 $p-1$ 阶的，在椭圆曲线算法中，将 Z_p^* 用 Z_p 上的随机椭圆曲线群代替。根据Hasse定理，这些群的阶分布在区间 $[p+1-2\sqrt{p},p+1+2\sqrt{p}]$ 上。若群的阶关于预先选定的界是光滑的，则算法有很高的概率得到 n 的非平凡因子；若群的阶不光滑，可能导致算法失败，可以更换椭圆曲线群重新执行算法。这一算法的时间复杂度为 $O(\exp((\sqrt{2}+o(1))(\ln p)^{1/2}(\ln\ln p)^{1/2}))$。若令 $L_q[a,c]=O(\exp((c+o(1))(\ln q)^a(\ln\ln q)^{a}))$，则上式可简记为 $L_n[1/2,\sqrt{2}]$。

2.3 平方同余方法(连分数算法，Dixon算法，二次筛法，数域筛法)

因子分解问题的另一类解法源于费马于1643年提出的方法。若一个奇数 $n=uv$，$u\leqslant v$，令 $x=(u+v)/2$，$y=(v-u)/2$，则 $n=x^2-y^2$。费马方法的主要思想就是寻找这样的 x,y。19世纪20年代M. Kraitchik推广了费马的想法，不要求 x^2-y^2 和 n 相等，而只要求它是 n 的倍数，即满足同余式

$$x^2\equiv y^2\pmod n,\ x\neq y\pmod n \qquad (*)$$

基于这一思想，1931年D. H. Lehmer和R. E. Powers提出了使用连分数分解因子的方法[10]，在1975年，M. A. Morrison和J. Brillhart发表了他们基于上述方法构造的适于计算机应用的算法[11]。

具体做法是对一个绝对值较小的整数 r，找到满足 $x^2\equiv r\pmod n$ 的 x 的值。若对某个 k,d 以及一个绝对值较小的 r，有 $x^2=r+knd^2$，则分式 x/d 是 $\sqrt{kn}$ 的一个近似；反之，若 x/d 是 $\sqrt{kn}$ 的一个较好的近似，则 $|x^2-knd^2|$ 将很小。于是通过 $\sqrt{kn}$ 的连分数展开，再利用二次无理式的连分数的性质，导出同余式 $x^2\equiv(-1)^{e_0}p_1^{e_1}p_2^{e_2}\cdots p_m^{e_m}\pmod n$ 的解。若找到该式的 $m+2$ 个解，则一些由 $e_0,e_1,\cdots,e_m$ 构成的向量，模2后线性相关。由此可找到(*)的解。这一算法的时间复杂度为 $L_n[1/2,\sqrt{2}]$。

1981年，J. D. Dixon提出了另一种新的整数因子分解算法[12]，同样以M. Kraitchik的思想为基础。其主要步骤如下：首先取前 t 个素数组成集合 $S=\{p_1,p_2,\cdots,p_t\}$，称为因子基。之后寻找满足如下条件的数对 (a_i,b_i)：$a_i^2\equiv b_i\pmod n$，且 b_i 是 p_t 光滑的。b_i 可表示为 $b_i=\prod_{j=1}^{t}p_j^{e_{ij}}$，则只要 e_{ij} $(j=1,2,\cdots,t)$ 均为偶数，b_i 即为完全平方。为此，类似连分数算法，令 $v_i=(v_{i1},v_{i2},\cdots,v_{it})$，其中 $v_{ij}=e_{ij}\bmod 2$，则找到 $t+1$ 个数对 (a_i,b_i)，产生的 $v_1,v_2,\cdots,v_{t+1}$ 中必有一组是线性相关的，由此即可得到 b_i 为完全平方的数对 (a_i,b_i)，从而可能到(*)的解。这里还需要确定产生数对 (a_i,b_i) 的方法，Dixon算法是简单地不断随机地选择 a_i，由 $b_i=a_i^2\bmod n$ 得到 b_i，再以因子基 S 中的素数试除，以确定 b_i 是否 p_t 光滑。而事实上，连分数算法可以看作Dixon算法的改进，通过 $\sqrt{kn}$ 的连分数展开求得 (a_i,b_i)。Dixon算法的时间复杂度为 $L_n[1/2,2\sqrt{2}]$。

1981年，C. Pomerance提出了二次筛法(QS)[13]，其主要想法和Dixon算法类似，但对数对 (a_i,b_i) 的选择方法作了重要的改进。

令 $m=\lfloor\sqrt{n}\rfloor$，考虑多项式 $Q(x)=(x+m)^2-n\approx x^2+2mx$，取 $a_i=x+m,b_i=Q(x)$，则 $a^2=(x+m)^2\equiv b_i\pmod n$，而要使 b_i 为 p_t 光滑的，就要取绝对值较小的 x，从而使 $Q(x)$ 较小，让 b_i 有较大的可能性是 p_t 光滑的。同时，若素数 p 整除 b_i，则 $(x+m)^2\equiv n\pmod p$，从而因子基只需包含 $(n/p)=1$ 的素数 p，其中 (n/p) 为勒让德符号。而因子基中元素的个数 $t\approx L_n[1/2,1/2]$ 时，算法有较高的效率，其时间复杂度为 $L_n[1/2,1]$，在数域筛法出现之前，二次筛法被认为是最有效的通用大数分解算法，从而被大量应用于实际的大数分解。

二次筛法有多种优化的方法，使用多重多项式和双大素数是其中较重要的两个。多重多项式二次筛选法(MPQS)用多个形如 $ax^2+2bx+c$ 的多项式代替 $Q(x)$，从而使每个多项式的筛选区间远小于使用单个的 $Q(x)$，从而提高算法的效率。在此基础上，又可以构造双大素数MPQS(PPMPQS)，即在得到不是 p_t 光滑的 b_i 时，不是将其立刻丢弃，而用一种机制有选择地将两个大素数加入因子基中，从而得到更多满足条件的 (a_i,b_i)，以更快地求得 n 的因子。自初始化的二次筛法(SIQS)[14]被提出之后，双大素数SIQS(PPSIQS)被认为在

一般情况下具有比 PPMPQS 更高的效率。

1988 年，J. M. Pollard 提出了一种新的技术，并与 A. K. Lenstra 等人共同发表了这种被称为数域筛法（NFS）的新算法[15]。该算法还是以求解（*）为目标，用不可约多项式的根 α 生成环 $Z[\alpha]$，然后利用环的性质以及 $Z[\alpha]$ 到 Z/nZ 的同态来构造（*）的解。该算法有两个不同版本：特殊数域筛法（SNFS）适用于分解形如 r^e-s 的整数，其中 r 和 s 都较小，其时间复杂度为 $L_n[1/3,1.526]$；而一般数域筛法（GNFS）可用于分解任意大整数，其时间复杂度为 $L_n[1/3,1.923]$。数域筛法被认为是目前最有效的超大整数分解算法，大约在 $n>10^{112}$ 之后，其效率可超过二次筛法[16]。

3 整数分解的量子算法：Shor 算法

上面介绍的传统算法的时间复杂度均是指数级，目前最优的复杂度为 $L_n[1/3,c]$。对于最优的传统算法，当待分解整数位数超过 100 时，计算量仍然是十分惊人的。如当整数为 250 位，用 1600 个工作站协同计算约需 80 万年。更重要的是，这个巨大的时间消耗仅仅通过计算机运算速度的提高在短时间内是无法明显改善的。一切迹象提示我们，整数分解问题连同许多同类问题可能在经典计算机上根本无法找到有效的解决办法。尽管上述理论还没有得到系统的证明，但包括数学家和计算机科学家在内的许多人愿意相信这类被称为 BQP 的问题等价类是客观存在的。

为了有效解决 BQP 问题，量子计算的概念与理论首先于 1982 年被物理学家 Richard Feynman 提出[17]，20 世纪末期，真正的量子计算设备已经在实验室被建造出来。

量子计算究竟是否能够完全解决 BQP 问题尚无定论，但理论和实验都表明，量子计算对于整数分解问题确实是存在比传统计算机更为有效的算法。量子计算可能超越经典计算的途径目前被认为是“量子超并行性（Quantum Parallelism）”。举例而言，在经典计算机中，信息单元用一个二进制位表示（称为“比特”，bit），在任意给定时刻，它所能表示的值要么为“0”要么为“1”。而在量子计算机中，信息单元被称为“量子位”，即 Qubit，它除了可以表示“0”和“1”以外，还可处于一种“0”和“1”之间的任意线性叠加态（State of Superposition），正是这种在传统计算中不存在的“中间态”，大大提高了量子计算的效率。在传统计算机中，一个 10 位的整数变量一次只能表示 0 到 1023 之间的某个数值，在进行函数运算时得到的结果也只能是对应于这一个特定输入的解；而一个具有 10 个量子位的整数量子变量一次可表示 0 到 1023 中所有的数值，一次函数运算也可将所有的这 1024 个值全部求出。虽然量子计算并不允许将所有的解一一列举，但它允许我们根据需要特定地加强某一个解输出的概率，从而轻易地得到隐含在一堆数据中的特定的解。这样就指数级的提高了运算的效率。

现已证明，通用量子计算对于上述的隐含子群问题（Hidden Sub-group Problem）可以指数级加速，而整数分解问题可以由数论理论化为寻找函数的周期的问题——这是典型的隐含子群问题。因此，利用量子超并行性进行快速因子分解的量子算法——Shor 算法诞生了。

算法先随机选择一个与 n 互质的正整数 m，考虑函数 $f(x)=m^x \bmod n$，使 $f(r)=1$ 的最小的正整数 r 就是该函数的周期。Shor 证明了，至少有 1/2 的概率使得 r 为偶数，且 $m^{r/2}\neq -1(\bmod n)$，而此时 $\gcd(x^{r/2}-1,n)$ 和 $\gcd(x^{r/2}+1,n)$ 中必有一个是 n 的非平凡因子。这样整数分解问题就转化成了求函数 $f(x)$ 的周期 r。这里要使用量子快速傅立叶变换（QFFT）算法：设 k 个量子位的量子寄存器的输入态 $|\psi_i\rangle=\sum_{i=\underbrace{00\cdots0}_{k}}^{\underbrace{11\cdots1}_{k}} f(x)|x\rangle$，则其 QFFT 为 $|\psi_i\rangle=\sum_{i=\underbrace{00\cdots0}_{k}}^{\underbrace{11\cdots1}_{k}}[2^{-k/2}\sum e^{2\pi ixx'}]f(x')|x\rangle$。事实上 QFFT 的结果就是将量子态前面的叠加系数变为原叠加系数的离散傅立叶变换。上述变换可以通过 $O(k^2)$ 个操作完成。通过由 QFFT 构造的相位估计算法，再利用连分数的性质，能够以较高的概率求得周期 r。虽然这一概率随着 n 的二进制长度的增加而减小，但速度只是多项式级的。因此，总体上仍然只需要多项式级的期望时间来得到 n 的因子。事实上整个算法的时间复杂度为 $O(\log(n)^3)$。

4 传统密码学所面临的挑战

通过上述的对比我们可以看到，在不久的将来，量子计算理论和量子计算机的进步将直接导致传统密码学理论基石的坍塌，由于 RSA 公钥密码体系目前应用极为广泛，几乎所有的电子商务、绝密电子邮件和银行系统都依赖于不同长度的 RSA 系统来保障安全。因此 RSA 体系的崩溃是我们最不愿意看到的。

目前，经典算法对 RSA 模数的分解已经有相当高的效率，使用 GNFS 算法，140 位十进制 RSA 模数在 1999 年被成功分解，计算量达到 2000 MIPS-years（即每秒百万次的处理器运行 2000 年）。对于这样的破解方式，主要的应对方法就是使用更大的 RSA 模数，现在 1024 位二进制的 RSA 密码系统已经被广泛使用，从而获得足够的安全性。

然而，随着量子计算机和量子算法的发展，在可预见的将来，即使通过继续增大规模，RSA 密码体系也无法再提供更高的安全性，因为 GNFS 算法的复杂度为 $L_n[1/3,1.923]$，随着待分解整数长度的增加而指数级的增加，而量子分解算法 Shor 算法则不同，其复杂度为 $O(\log(n)^3)$，只是待分解整数长度的多项式级别。初步估计，在量子计算机上运行 Shor 算法，按每秒一百万次操作计算，分解 1024 位二进制数大约只需 18 分钟，而若用 GNFS 算法来分解，则其计算量相当于 RSA140 的 4900 万倍，即 980 亿 MIPS-years。

我们将不再长期处于传统计算机时代而即将进入全新的量子计算时代。在这种计算体系彻底转变的可能性面前，未雨绸缪显得尤为重要。按照目前的理论，要保证信息的绝对安全必须依靠量子计算机和完善的量子加密算法以及量子密钥分配系统，这种全新的量子密码体系具有不可窃听、超密度编码加密和密钥加密传输等诸多传统密码学无法比拟的优点。现在对这几类问题的研究正在进行并已经取得部分很有价值的结果。我们相信，在不久的将来，传统密码学将被量子密码学部分甚至全部取代，那时的信息安全将坚实地建立在物理体系而不仅是逻辑体系之上，也许目前所面临的一系列信息泄露的危机将会得到有效的避免。

结束语 本文介绍了整数因子分解的重要意义，并综述了近代提出的一系列经典整数分解算法，对每类算法给出时间复杂度并进行了比较。最后，通过引入量子计算的原理介绍了量子整数因子分解算法——Shor 算法，并由此讨论了传统密码学所面临的巨大挑战。

参 考 文 献

[1] Rivest R L, Shamir A, Adleman L. A Method for Obtaining Dig-

ital Signatures and Public-Key Cryptosystems. Communications of the ACM, 1978, 21(2): 120-126

[2] Nielsen M A, Chuang I L. Quantum Computation and Quantum Information. Cambridge University Press, 2000

[3] Wang X, Yin Y L, Yu H. Finding Collisions in the Full SHA-1. CRYPTO, 005

[4] Wang X, Yu H. How to Break MD5 and Other Hash Functions. EUROCRYPT, 2005

[5] Pollard J M. A Monte Carlo method for factorization. BIT Numerical Mathematics, 1975, 15(3): 331-334

[6] Brent R P. An Improved Monte Carlo Factorization Algorithm. BIT, 1980, 20: 176-184

[7] Pollard J M. Theorems of Factorization and Primality Testing. Cambridge Philosophical Society, 1974, 76: 521-528

[8] Williams H C. A p+1 method of factoring. Mathematics of Computation, 1982, 39: 225-234

[9] Lenstra H W Jr. Factoring integers with elliptic curves. Annals of Mathematics, 1987, 126(2): 649-673

[10] Lehmer D H, Powers R E. On Factoring Large Numbers. Bulletin of the American Mathematical Society, 1931, 37: 770-776

[11] Morrison M A, Brillhart J. A Method of Factoring and the Factorization of F7. Mathematics of Computation, 1975, 29 (129): 183-205

[12] Dixon J D. Asymptotically fast factorization of integers. Mathematics of Computation, 1981, 36: 255-260

[13] Pomerance C. The Quadratic Sieve Factoring Algorithm. EUROCRYPT1984 // Pomerance C, ed. A Tale of Two Sieves. Not. Amer. Math. Soc., 1996, 43: 1473-1485

[14] Contini S. Factoring integers with the self-initializing quadratic sieve. Masters thesis. University of Georgia, 1997

[15] Lenstra A K, Lenstra H W Jr, Manasse M S, et al. The Number Field Sieve // ACM Symposium on Theory of Computing. 1990: 564-572

[16] Knuth D E. The Art of Computer Programming. Third Edition. Addison Wesley, 1998, 2

[17] Feynman R P. Simulating physics with computers. International Journal of Theoretical Physics, 1982, 21: 467-488

[18] Shor P W. Algorithms for quantum computation: discrete logarithms and factoring. New Mexico: IEEE Computer Society Press, 1994: 124-134

[19] Cormen T H, Leiserson C E, Rivest R L, et al. Introduction to Algorithms, Second Edition. The MIT Press, 2001

[20] Menezes A J, van Oorschot P C, Vanstone S A. Handbook of Applied Cryptography. CRC Press, 1997

[21] Brent R P. Recent Progress and Prospects for Integer Factorisation Algorithms. COCOON, 2000

(上接第 5 页)

[43] Akyildiz I F, Kasimoglu I H. Wireless sensor and actor networks: research challenges. Ad Hoc Networks, 2004, 2(4): 351-367

[44] Chong C Y, Kumar S. Sensor networks: evolution, opportunities, and challenges // Proceedings of the IEEE. 2003, 91(8): 1247-1256

[45] Estrin D, Govindan R, Heiderdmann J, et al. Next century challenges: Scalable coordination in sensor networks // Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking. Washington, USA, 1999: 263-270

[46] Smith D, Ma L, Ryan N. Acoustic environment as an indicator of social and physical context. Personal and Ubiquitous Computing, 2006, 10(4): 241-254

[47] Abowd G D, Mynatt E D. Charting past, present and future research on ubiquitous computing. ACM Transactions on Computer-Human Interaction, 2000, 7(1): 29-58

[48] Castro P, Muntz R. Managing context data for smart spaces. IEEE Personal Communications, 2000, 7(5): 44-46

[49] 岳玮宁, 王悦, 汪国平, 等. 基于上下文感知的智能交互系统模型. 计算机辅助设计与图形学学报, 2005, 17(1): 74-79

[50] Chen H, Finin T, Joshi A, et al. Intelligent agents meet the semantic web in smart spaces. IEEE Internet Computing, 2004, 8(6): 69-79

[51] Voida S, Mynatt E D, MacIntyre B, et al. Integrating virtual and physical context to support knowledge workers. IEEE Pervasive Computing, 2002, 1(3): 73-79

[52] Saba D, Mukherjee K. Pervasive computing: A paradigm for the 21st century. IEEE Computer, 2003, 36(3): 25-31

[53] Yau S S, Karim F, Wang Yu, et al. Reconfigurable context-sensitive middleware for pervasive computing. IEEE Pervasive Computing, 2002, 1(3): 33-40

[54] Yu Y, Krishnamachari B, Prasanna V K. Issues in designing middleware forwireless sensor networks. IEEE Network, 2004, 18(1): 15-21

[55] Soldatos J, Pandis I, Stamatis K, et al. Agent based middleware infrastructure for autonomous context-aware ubiquitous computing services. Computer Communications, 2007, 30(3): 577-591

[56] Romn M, Hess C, Cerqueira R, et al. A middleware infrastructure for active spaces. IEEE Pervasive Computing, 2002, 1(4): 74-83

[57] Romer K, Kasten O, Mattern F. Middleware challenges for wireless sensor networks. ACM SIGMOBILE Mobile Computing and Communications Review, 2002, 6(4): 59-61

[58] Ollero A, Boverie S, Goodall R, et al. Mechatronics, robotics and components for automation and control. IFAC milestone report. Annual Reviews in Control, 2006, 30 (1): 41-54

[59] Rezgui A, Eltoweissy M. Service-oriented sensor-actuator networks: promises, challenges, and the road ahead. Computer Communications (to be published)

[60] 徐振阳, 窦文华. 无线传感反应网络综述. 计算机科学, 2005, 32(9): 26-30

[61] Liu J, Chu M, Liu J, et al. State-centric programming for sensor-actuator network systems. Pervasive Computing, 2003, 2(4): 50-62

[62] Pietro R D, Mancini L V, Jajodia S. Providing secrecy in key management protocols for large wireless sensors networks. Ad Hoc Networks, 2003, 1(4): 455-468

[63] Kagal L, Finin T, Joshi A. Trust-based security in pervasive computing environments. IEEE Computer, 2001, 34(12): 154-157

[64] Perrig A, Stankovic J, Wagner D. Security in wireless sensor networks. Communications of the ACM, 2004, 47(6): 53-57

[65] Dritsas S, Gritzalis D, Lambrinoudakis C. Protecting privacy and anonymity in pervasive computing: trends and perspectives. Telematics and Informatics, 2006, 23(3): 196-210