云南大学学报（自然科学版），2011，33（1）：22～26　　CN 53－1045/N　ISSN 0258－7971
Journal of Yunnan University　　http://www.yndxxb.ynu.edu.cn

基于多核的批处理 RSA 的并行加速方法*[1]

李云飞[1]，柳　青[2,3]，李　彤[2,3]，周保林[1]，彭　华[1]
（1. 云南大学 信息学院，云南 昆明　650091；2. 云南大学 软件学院，云南 昆明　650091；
3. 云南省软件工程重点实验室，云南 昆明 650091）

摘要：为了改善 RSA 算法解密和签名的性能，Fiat 提出了 batch RSA 算法，但效果并不显著. 针对现有计算机多核的特点，对 batch RSA 算法进行并行优化，使其在解密和签名时的速度得到大幅度提升，实验表明并行优化后平均加速比可达到 4.75.

关键词：batch RSA；加速；并行；多核
中图分类号：TP 309　　**文献标识码**：A　　**文章编号**：0258－7971(2011)01－0022－05

RSA 是目前应用最为广泛的公钥密码算法[1]，它不仅可以用来加密，还可以用来进行数字签名，在安全 Web 传输，E－mail 和某些无线设备中被大量使用. 由于 RSA 计算模正整数次幂需要耗费大量的计算时间，从而降低了 RSA 密码算法的执行效率. 特别是进行解密和数字签名时，要进行大数的模幂运算，会耗费更多的计算时间，所以如何提升 RSA 密码算法的解密性能并使其尽可能少消耗资源是密码学界一直研究的问题. 多年来，出现了多种 RSA 的改进算法来加速 RSA 的解密和数字签名过程，如：batch RSA[2－3]，Multi－factor RSA[4] 和 Rebalanced RSA[4] 等，这些改进算法试图通过不同的方法来提高 RSA 的解密和签名速度.

以上这些改进算法被提出来时，所基于的硬件设备都是单核处理器. 从 2005 年开始，通用多核处理器开始进入市场，这种处理器通过多核架构中的一整套 CPU 的并行工作来获取更高的性能[5]. 多核处理器包含多个处理器单元，每个处理器单元具有独立的缓存和 I/O 等资源，可以同时运行. 各处理单元通过内部总线互连，具有很高的通信带宽. 从理论上讲，如果不考虑软件处理带来的多个处理器单元同步及通信的开销，N 核处理器系统的处理能力相当于同频率的单核处理器的 N 倍[3]. 因此多核处理器使系统性能大幅提升. 根据当前计算机市场，多核处理器已成为计算机处理器的主流发展方向. 多核处理器可以构建具有并行处理能力的平台，能有效提高处理器的计算能力. 许多安全系统都是以 PC 作为硬件平台，研究如何利用多核处理器提升安全系统的处理性能，具有重要的现实意义和使用价值. 本文以 batch RSA[3] 为出发点，首先对该 RSA 改进算法进行分析. 然后结合当前硬件设备以多核为主的特点，充分发挥多核设备的优点，提出了基于多核的 batch RSA 并行处理模型，接着对模型进行了实现，最后进行测试. 测试结果显示，在双核处理平台上，并行 batch RSA 解密性能得到较大的提升，实验结果显示，相对于并行前的标准 RSA 算法，并行优化后的 batch RSA 算法解密平均加速比可达 4.75.

1　batch RSA 算法

Fiat 发现若使用小公钥指数 e_1 和 e_2 对 2 个明文进行 RSA 加密处理，对得到的 2 个密文进行解密

* **收稿日期**：2010－03－23
基金项目：云南大学中青年骨干教师培养计划资助项目(21132014)；国家自然科学基金资助项目(60963007)；云南省自然科学基金资助项目(2007F008M)；云南省软件工程重点实验室开放基金资助项目(2010KS01).
作者简介：李云飞(1986－)，男，云南人(白族)，硕士，主要从事密码学与信息安全方面的研究.
通讯作者：柳　青(1963－)，男，云南人，教授，主要从事软件工程、信息安全方面的研究.

处理时,只需消耗解密 1 个密文的代价[3]. 假设 v_1 是通过公钥〈N,3〉加密得到的密文;v_2 是通过公钥〈N,5〉加密得到的密文. 若要对 v_1 和 v_2 进行解密,必须要计算 $v_1^{1/3}$ 和 $v_2^{1/5}$ mod N. Fiat 发现通过设 $A = (v_1^5 \cdot v_2^3)^{1/15}$ 并进行以下计算可得到 $v_1^{1/3}$ 和 $v_2^{1/5}$ 的值:

$$V_1^{1/3} = \frac{A^{10}}{v_1^3 \cdot v_2^2} \text{ 和 } V_2^{1/5} = \frac{A^6}{v_1^2 \cdot v_2}.$$

这样只需要计算一个 15 次方根,再加上一些模乘和模除运算就可以同时解密 v_1 和 v_2. 以上是解密 2 个密文的情况,现在扩展到对 b 个密文进行成批解密:设有 b 个不同且两两互素的公钥加密指数 e_1,…,e_b,它们共用相同的模 N. 同时存在 b 个明文 m_1,…,m_b,并分别用对应的公钥指数 e_i 加密,得到密文 v_1,…,v_b,获得密文通过:$v_i = m_i^{e_i}$ mod N. 批处理的过程可以用1棵拥有 b 个叶子节点的完全二叉树来实现,且该树内部节点必须拥有左右孩子. 树中的每个节点包含 2 个值:1 个是公钥指数值,1 个是密文计算值,本文中分别用大写字母 E 和小写字母 v 来表示. 如图 1 所示,每个节点上有 2 个值,其中数字内容为公钥指数值,小写字母表示密文计算值. batch RSA 算法包括3 个阶段: batch 树构造,指数计算和 batch 树分解.

(1) batch 树构造　该阶段主要目的是将各个独立的密文消息 v_i 结合成为 1 个整体,在树的根部得到:$V = \prod_{i=1}^{b} v_i^{e/e_i}$,其中 $e = \prod_{i=1}^{b} e_i$. batch 树构造的过程是一个自底向上的过程,首先用 b 个公钥指数来标注树的 b 个叶子节点,这里使用大写字母 E 来表示公钥指数:$E \leftarrow e_i$,其中内部节点的公钥指数的值是其左右孩子公钥指数值的乘积:$E \leftarrow E_L \cdot E_R$. 这样 batch 树的根节点的公钥指数 E 值是 b 个公钥指数的乘积,即 $E \leftarrow e = \prod_{j=1}^{b} e_i$. 接下来把 b 个密文信息 v_i 作为相应公钥指数叶子节点的输入,对密文信息 v 进行计算:$v \leftarrow v_L^{E_R} \cdot v_R^{E_L}$. 图1 显示了 $e_1 = 3, e_2 = 5, e_3 = 7$ 和 $e_4 = 11$ 时 4 个公钥指数进行 bacth 树构造得到 E 和 v 的过程.

(2) 指数计算　构造 batch 树阶段完成之后,根节点的密文计算值为 $V = \prod_{i=1}^{b} v_i^{e/e_i}$. 在指数计算阶段,对 V 的 e 次方根进行计算. 经过这一阶段的处理后,得到:$m = V^{\frac{1}{e}} = \prod_{i=1}^{b} v_i^{1/e_i}$.

(3) batch 树分解　该阶段是把 m 分解成为左右孩子的乘积并且最终在 batch 树的叶子节点上得到各个密文 v 的明文. batch 树的分解是一个自顶向下的过程,具体分解的方法是:首先,在每个内部节点选择1 个 X,使其满足:$X = 0 (\text{mod } E_L)$ 和 $X = 1 (\text{mod } E_R)$. X 的构造使用中国剩余定理[7],并定义 2 个变量 X_L 和 X_R,它们的值为:$X_L = X/E_L$ 和 $X_R = (X-1)/E_R$,式子中的2个除法都是整数上的除法. 如 Fiat 所总结的,此时 X, X_L 和 X_R 对于每个内部节点有:$m^X = v_L^{X_L} \cdot v_R^{X_R} \cdot m_R$[4] 可以通过以下方式进行下分解:$m_R = m^X/(v_L^{X_L} \cdot v_R^{X_R})$ 和 $m_L = m/m_R$. 通过这种方法不断迭代向下分解. 当分解过程结束时,每个叶子节点的明文 m 都被解密出来. 图 2 显示了 $e_1 = 3, e_2 = 5, e_3 = 7, e = 11$ 和 $m = V^{1/e} = v_1^{1/3} \cdot v_2^{1/5} \cdot v_3^{1/7} \cdot v_4^{1/11}$ 时,batch 树分解和最终得到 4 个明文的过程.

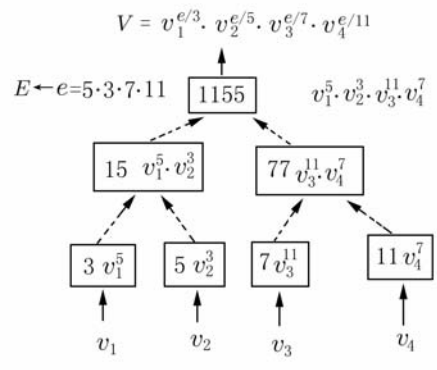

图 1　batch 树向上渗透过程

Fig. 1　The batch tree Percolate – UP

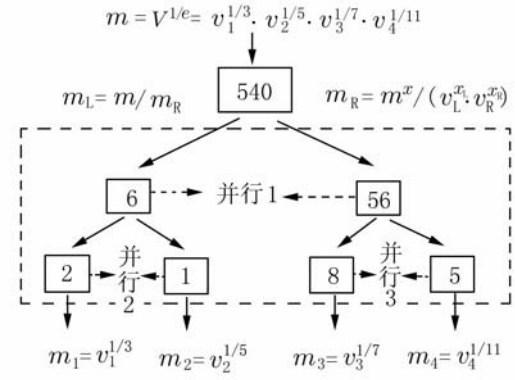

图 2　batch 树向下渗透过程和 batch RSA 算法内部并行处理示意图

Fig. 2　The batch tree Percolate – Down and decryption parallelism in the batch RSA algorithm

在 batch 解密的整个过程中,只使用了 1 次全指数模幂运算,替代了 b 次全指数模幂运算. 提升了 batch RSA 算法的解密性能. 接下来,本文依据当前硬件设备的多核特点,通过基于多核的并行优化的方法来提升 batch RSA 解密和数字签名的性能.

2 batch RSA 并行优化

本文希望通过基于多核的并行程序在较少的时间内解决较大问题. 但是,只有当问题具有可并行性时,即多个活动或任务能够在同一时间内同时执行时,它才能起作用[8]. 所以要设计一个并行解决方案,应当首先在问题领域内分析问题,以揭示可开发的并行性. 本文从 2 个方面上对 batch RSA 算法进行并行优化:① 算法自身的并行;② 多个算法整体上的并行,从 2 个方面设计 batch RSA 的并行优化模型.

2.1 batch RSA 并行优化模型 本文首先从寻找 batch RSA 算法的并行性入手,设计与其相适应的并行优化方案. 本文的第 2 部分对 batch RSA 算法的各个阶段进行了描述,现来划分算法的解密和加密阶段. 当前的各种加密系统或签名认证系统都是基于网络服务,所以都存在客户端和服务器端. 这些系统的服务器端在进行解密和数字签名时会耗费大量的计算资源,使得服务器负载过重. 为了尽可能减少服务器端的负载,在划分加密端和解密端时,应尽可能将解密端一些工作转移给客户端来完成. 在分析 batch RSA 算法的各个阶段之后,为了能让客户端多做一些工作,在各个公钥指数相对固定的前提下,客户端可以在完成加密操作之后,处理解密端一部分的模幂运算. 从而使得解密端负载降低,提升 RSA 算法的解密性能。图 2 显示了已将一部分解密操作转移到加密端的 batch 树的分解解密过程. 本节将由此入手,寻找 batch RSA 的并行性. 寻找并行性的第 1 步是将问题分解为多个能够并发执行的元素. 分解可以从 2 方面去考虑:数据分解和任务分解[8]. 数据分解集中于分析任务所需要的数据. batch RSA 解密阶段中最主要的数据是经过模幂计算处理后的 m 值,如图 2 所示的 batch 树的顶部输入,由于此时数据 m 是一个数据整体,不能划分,所以此时不能进行并行处理,但接下来数据 m 被分解为左右 2 个数据后,数据块之间是相互独立的,为以下功能分解奠定了基础,只有数据块能被相对独立地操作时,与数据相关的计算才能够高效地执行. 接下来进行任务分解,该分解是将问题看作一个指令流,指令流中的指令能够被分解为多个称为任务的序列,batch RSA 解密部分的任务是计算 $m_R = m^X/(v_L^{X_L} \cdot v_R^{X_R})$ 和 $m_L = m/m_R$. 数据 m 被分解后的数据 m_L 和 m_R 以及 V_L 和 v_R 并且各数据之间是相互独立的,加上独立的数据处理功能,所以该部分可实现并行. 并行处理模型要与硬件的核数和 batch 长度相匹配才能达到很好的效果. 图 2 显示了基于四核的 batch 长度为 4 的并行模型,其中虚线框中显示了可并行的 3 个地方. 根据双核的硬件设备,可以将 2 和 3 两个部分串行,只实现 1 部分的并行. 并行部分的确定可根据任务要求的计算量大小来确定,若任务本身要求计算量很大时,那么对这些任务进行并行优化才有意义.

2.2 多个 batch RSA 整体并行优化模型 以上是对 batch RSA 的算法自身的并行优化,本小节将对多个 batch RSA 算法进行解密处理的情形进行分析,设计出与其相适应的并行优化方案. 现对多个 batch RSA 解密处理的情形进行数据分解和任务分解分析,以寻找并行性. 多个 batch RSA 解密处理输入的数据是经过加密处理的密文,这些数据相互独立,是功能并行的基础. 对于功能分解,由于是用多个 batch RSA 来进行解密,所以存在很明显的并行性,多个 batch RSA 解密操作可以同时进行处理. 图 3 显示了硬件为双核时,batch RSA 长度为 4 的情况下,batch RSA 同时处理 8 个密文的整体并行模型.

对于以上 2 种模型,本文通过 OpenMP[9] 来实现并行优化, OpenMP 是一种面向共享内存以及分

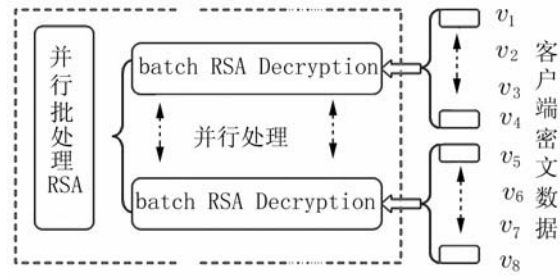

图 3 2 个 batch RSA 算法解密并行处理示意图

Fig. 3 Decryption parallelism in two batch RSA algorithms

布式共享内存的多线程并行编程语言和可用于编写可移植的多线程应用程序的 API,广泛应用于多核程序设计中. 对于 batch RSA 自身的并行优化,通过以上并行性分解分析可知,可以将 $m_R = m^X/(v_L^{X_L} \cdot v_R^{X_R})$ 和 $m_L = m/m_R$ 进行并行化,让每个线程处理 1 个操作,实现可以通过 OpenMP 的 parallel sections 语句. 对于整体 batch RSA 的并行可以让 1 个线程处理 1 个 batch RSA 解密,并行的数量可以根据硬件核数和 batch 的长度来确定. 在 OpenMP 并行化过程中,若并行数量较少时,可以通过 parallel sections 来进行并行优化,若并行数量较多时,则使用 parallel for 来进行优化. 与未优化前的 batch RSA 相比,该并行优化在很大程度上提高了解密和签名的效率,充分利用了硬件设备的多核特性.

3　性能分析

上述优化方案的实现使用了 OpenSSL(0.9.8k)库和 OpenMP. 硬件实验平台为:操作系统 Windows XP,处理器 Intel Pentium 双核 1.73 GHz,内存 1GB. 在实现过程中,选择的批处理长度为 4,实现了 batch RSA 的整体并行优化方案(PBRSA),同时也实现了标准 RSA 算法和 batch RSA(BRSA)算法,以及简单的 RSA 并行(PRSA),表 1 显示了解密 8 个密文数据时,各类算法解密时的耗时情况.

表 1　4 种类型的 RSA 算法解密 8 个密文耗费时间(ms)

Tab. 1 Decryption time for four RSA variants decrypting eight ciphertexts

算法类型	密钥长度 /bits				
	1 024	1 536	2 048	2 560	3 072
RSA	63	140	359	578	906
BRSA	31	63	125	172	266
PRSA	31	78	188	297	453
PBRSA	15	32	63	109	141

表 2 中显示了 3 种算法相对于标准 RSA 解密 8 个密文时的加速比,从表 2 中可以看出并行处理后的 batch RSA 算法是 3 种算法中加速比最高的算法,其解密时的平均加速比达到 4.75. 图 4 中分别显示了解密 8 个密文时的 batch RSA(BRSA),并行 RSA(PRSA)和并行 Batch RSA(PBRSA)的密钥长度与加速比的变化关系. 从图 4 中不难看出随着密钥长度的增大,3 种 RSA 算法的加速比呈上升趋势.

表 2　4 种类型的 RSA 算法相对于标准 RSA 算法的差异密加速比

Tab. 2 Decryption speedup to standard RSA algorithm for four RSA variants

算法类型	密钥长度 /bits						平均值
	1 024	1 280	1 536	2 048	2 560	3 072	
BRSA	2.0	1.65	2.22	2.87	3.36	3.40	2.58
PBRSA	4.20	2.52	4.37	5.69	5.30	6.42	4.75
PRSA	2.03	1.66	1.79	1.90	1.95	2.00	1.89

最大加速比为 PBRSA 在密钥长度为 3 072 时的 6.42. Batch RSA 的并行优化方案使得 RSA 数字签名和数据解密时的性能得到大幅度提升,充分发挥了硬件设备的双核处理器的处理能力,减少了处理时间和延迟,保证了高质量的安全服务.

4　结束语

通过对 batch RSA 算法解密的并行优化,使得 batch RSA 算法的解密性能获得较大提升. 在安全应用系统中,RSA 算法的解密和签名操作时非常耗费资源的,它限制了 RSA 安全系统的整体性能. 所以,通过设计良好的并行处理方案,充分发挥多核平台的强大处理能力,对提升整个安全系统的性能有非常重要的意义.

参考文献:

[1] RIVEST R, SHAMIR A, ALDEMAN L. A methoed for obtaining digitalsignatures and public - key cryptosystems[J]. Communications of the ACM, 1978, 21(2): 120-126.

[2] FIAT A. Batch RSA[M]. Berlin: Springer - Verlag, 1989.

[3] 齐芳,贾维嘉. SSL 握手协议中客户端平衡密钥交换算法[J]. 计算机工程与应用,2007,43(19):1-6.

[4] BONEH D, SHACHAM H. Fast variants of RSA[R]. RSA Laboratories Cryptobytes, 2002.

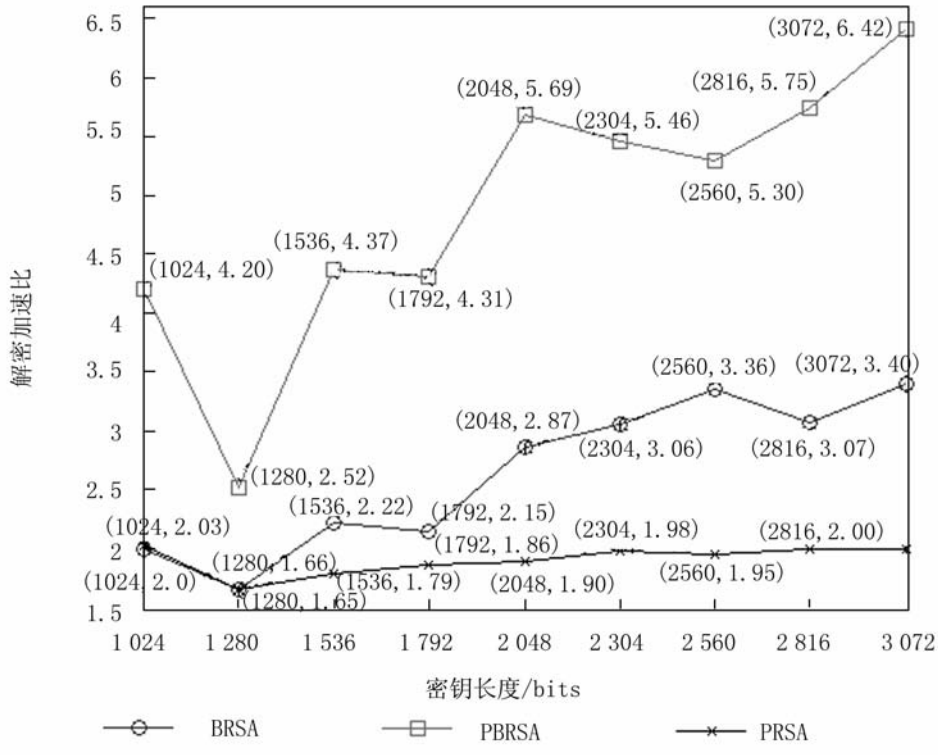

图 4　3 种类型 RSA 算法密钥长度和解密加速比的关系图

Fig. 4 Decryption speedup when varying key size for three RSA variants

[5] PAXSON V, SOMMER R. An architecture for exploiting multi – core processors to parallelize network intrusion prevention[C]//Proceedings of the IEEE Sarnoff Symposium, 2007: 1-7.

[6] 刘近光, 梁满贵. 多核多线程处理器的发展及其软件系统架构[J]. 微处理机, 2007, 1(2): 1-3.

[7] 闵嗣鹤, 严士健. 初等数学[M]. 2 版. 北京: 高等教育出版社, 2003.

[8] TIMOTHY G, BEVERLY A. 并行程序设计模式[M]. 敖富江, 译. 北京: 清华大学出版社, 2005.

[9] MICHAEL J. MPI 与 OpenMP 并行程序设计[M]. 陈文光, 武永卫, 译. 北京: 清华大学出版社, 2004.

Parallel accelerated method of batch RSA based on multi – core processor

LI Yun-fei[1], LIU Qing[2,3], LI Tong[2,3], ZHOU Bao-lin[1], PENG Hua[1]

(1. School of Information Science and Engineering, Yunnan University, Kunming 650091, China;
2. National Pilot School of Software, Yunnan University, Kunming 650091, China;
3. Key Laboratory in Software Engineering of Yunnan Province, Kunming 650091, China)

Abstract: Fiat had put forward the batch RSA to speed up the decryption and signing, but it wasn't very effecient. Considering the multi – core feature of present computers, this paper focused on the parallel optimization of batch RSA in order to further speed up decryption and signing. Experiments finally showed that the speedup can reach a factor of 4.75.

Key words: batch RSA; accelerate; parallel; multi – core