SSE2 与 AES-NI 在密码算法中的应用*

苗旭东[2]** 郑秀林[1,2] 李艳俊[1]

1. 北京电子科技学院信息安全系,北京 100070;

2. 西安电子科技大学通信工程学院,西安 710071

摘 要: 密码算法的实现效率是衡量一个密码算法好坏的重要指标。传统密码算法是基于比特(A5 算法)、字节(AES 算法)、32 位字(IDEA)设计的,软件实现速度相对较慢,而在最近的 CAESAR 竞赛中,基于快速指令集设计的一些算法,例如 MORUS、AEGIS,它们的软件实现速度是非常快的。本文选择了两种非常有代表性的指令集 SSE2 与 AES-NI 进行研究,研究这两种指令集在密码算法中的使用方法,以及这些指令集应用到密码算法中会提升效率,通过对这两种指令集的研究,指出了使用指令集在密码算法中的优势。

关键词: SSE2; AES-NI; 指令集

中图分类号: TN918.1 **文献标识码:** A **文章编号:** 1672-464X(2016)2-12-05

引 言

本文的讨论范围是基于软件实现的密码算法,目标实现平台是当前的 CPU 处理器。设计密码算法时,一般都会考虑密码算法的实现效率问题,密码算法如何设计才能实现起来更快、占用内存资源更少?这需要考虑目标实现平台所擅长执行的操作。传统的对称密码是基于比特(A5 算法)、字节(AES 算法)或 32 位字(IDEA 算法)设计的,一般来说,基于字节设计的密码算法要比基于比特设计的密码算法软件实现更有效,而基于字设计的密码算法要比基于字节设计的密码算法软件实现更有效,而目前的 CPU 处理器上集成有很多处理更长数据块的操作指令,如 SSE2 指令集(可处理 128 位的数据)、AVX2 指令集(可处理 256 位的数据)和 AES 新指令集 AES-NI(处理 128 位的数据),这些指令集可以用到密码算法的设计和实现中,例如,目前正在进行的 CAESAR 竞赛[1]中基于 SSE2 指令集设计的 MORUS-640 算法[2]的软件实现效率可达 1.11cycle/byte,基于 AVX2 指令集设计的 MORUS-1280 算法[2]的软件实现效率可达 0.69cycle/byte,基于 AES-NI 实现的 AEGIS-128 算法[3]的软件实现效率可达 0.61cycle/byte。可以看出,使用指令集设计的密码算法实现效率很高,这可能是未来几年设计密码算法的一个趋势。本文以 SSE2 指令集和 AES-NI 为例,介绍这两种指令集的功能和使用方法,测试它们的实现效率,并分析了在密码算法中使用这两种指令后可提升的效率。

1 SSE2 指令集

SSE2 指令集[4]是在 2000 年出现的奔腾 4 处理器、至强处理器中引入的,目前的 CPU 处理器中几乎都集成有 SSE2 指令集,它主要利用的是 SIMD(single-instruction multiple-data)模型以

* 基金项目:中央高校基本科研业务费资助(项目编号:328201531)

** 作者简介:苗旭东(1989-),男,汉族,硕士研究生,主要从事密码学领域的研究。

提升数据操作的速度，即执行一条指令并行处理多组数据。SSE2 指令最大可操作 128 位的整数，在密码算法中，通常都是对整数的操作，这也是本文研究 SSE2 指令集的原因。

1.1　几条典型 SSE2 指令的功能与使用

在对称密码算法中，通常选择运算速度快、扩散效果好、混乱性好的操作，比较常用的几种操作是：异或操作、移位操作、逻辑与操作、模加运算。本文会从 SSE2 指令集中选择一些与这几种操作相关的指令，并对它们的实现性能进行研究。

表 1 列出了 4 条非常具有代表性的 SSE2 指令，并列出了每条指令的名称，大致描述了每条指令实现的功能，同时给出了每条指令对应的 C 语言函数[5]。

表 1　SSE2 指令

指令名称	实现的功能	C 函数
PXOR	128 位数异或	__m128i_mm_xor_si128 (__m128i a, __m128i b)
PAND	128 位数逻辑与	__m128i_mm_and_si128 (__m128i a, __m128i b)
PSLLDQ	128 位数左移	__m128i_mm_slli_si128 (__m128i a, intimm8)
PADDD	模 2^{32} 加法运算	__m128i_mm_add_epi32 (__m128i a, __m128i b)

其中，PXOR 指令是对 128 位的整数 a 与 b 做异或操作、PAND 指令是对 128 位的整数 a 与 b 做逻辑与操作，PSLLDQ 指令是对 128 位的整数 a 左移 8 × imm8 位，PADDD 指令是将 128 位的整数 a 与 b 分别分成 4 个 32 位的字，对应字分别做模 2^{32} 加法运算。

1.2　SSE2 指令集的实现效率

了解这 4 条 SSE2 指令的实现功能后，下面的实验将测试这 4 条 SSE2 指令的实现效率，为了与普通指令的实现效率做对比，在相同平台下也测试了与 SSE2 指令对应的普通指令的实现速度。

实验环境：在 Intel (R) Core (TM) 2 Duo E8400 @ 3.0GHz 的处理器上执行，32 位 Windows7 操作系统，编译器为 vs2010。未使用任何加速技术，编程用标准 C 语言实现。C 语言测试 SSE2 指令需要加头文件 # include " emmintrin. h"。

表 2 列出了测试所得数据，表中给出执行一条 SSE2 指令平均需要的时钟周期数，并且在紧接着 SSE2 指令测试数据的下一行给出了与之相应普通指令的耗时，第三列对比了 SSE2 指令与普通指令的实现效率。效率比较时，需要注意的是，一条 SSE2 指令共操作 128 位的数据，与之相应的普通指令只操作 32 位的数，相当于，一条 SSE2 指令执行的操作至少需要普通指令执行 4 次才能完成，因此，普通指令的耗时需要乘以 4 再与 SSE2 指令的实现效率相比较。

表 2　SSE2 指令与普通指令耗时比较

指令名称	耗时(cycles)	效率比较
PXOR 指令	11	4* 6/11 ≈2.2
32 位异或“^”指令	6	
PAND 指令	11	2.2
32 位与“&”指令	6	
PSLLDQ 指令	11	2.2
32 位左移 “《”指令	6	
PADDD 指令	12	2
32 位模加“ + ”运算	6	

从上表中的测试数据可知，使用 SSE2 指令会比使用普通指令快一倍多，在密码算法的设计中，提升一倍的效率是相当可观的。因此，在密码算法的设计过程中，在不降低安全性的前提条件下，考虑使用 SSE2 指令集来提升密码算法的效率还是很有必要的。

2　AES 新指令集 AES-NI

高级加密标准(AES) 是美国联邦政府采用

的对称加密标准，从安全性与实现效率来看，AES 被广泛认为是安全高效的。因此被众多政府、企业等作为加密标准。并且，有很多密码算法是基于 AES 设计的，如消息认证码 CMAC、PMAC，随机数生成算法 CTR-DRBG，以及认证加密算法 AES-GCM 与 OCB。在最近的 CAESAR 竞赛中很多设计者提交的算法都是基于 AES 设计的认证加密算法。因此，研究如何提升 AES 算法的执行速度尤为重要。

2.1 AES-NI 功能与使用

AES 新指令集简称 AES-NI[6]，从 2010 年起，英特尔推出 32 纳米微体系结构、代号为 westmere 的处理器中集成有 AES 新指令集。AES-NI 共有六条指令，四条指令用来加解密（AESENC，AESENCLAST，AESDEC，AESDELAST），两条指令用来扩展密钥（AESIMC，AESKEYGENASSIST）。表 3 列出了 AES-NI 的 6 条指令，并描述了每条指令实现的功能。

表 3 AES-NI 对应的功能

指令名称	功能
AESENC	执行一轮 AES 加密
AESENCLAST	执行最后一轮的 AES 加密
AESDEC	执行一轮 AES 解密
AESDELAST	执行最后一轮的 AES 解密
AESIMC	相当于 AES 的逆向列混合操作（InvMixColumns）
AESKEYGENASSIST	用来辅助密钥扩展操作

软件实现密码算法时，通常使用 C 语言编写，表 4 列出了 6 条指令对应的 C 函数[2]，需要说明的是，若不考虑 AES 的密钥扩展，表 4 中列出的前四条指令就可实现 AES 的加解密，而 AES 的密钥扩展算法，需要用到后两条指令，文献[6]中介绍了使用后两条指令扩展密钥的具体方法。

表 4 AES-NI 对应的 C 函数

指令名称	C 函数
AESENC	__m128i _mm_aesenc_si128 (__m128i a, __m128i RoundKey)
AESENCLAST	__m128i _mm_aesenclast_si128 (__m128i a, __m128i RoundKey)
AESDEC	__m128i _mm_aesdec_si128 (__m128i a, __m128i RoundKey)
AESDELAST	__m128i _mm_aesenclast_si128 (__m128i a, __m128i RoundKey)
AESIMC	__m128i _mm_aesimc_si128 (__m128i a)
AESKEYGENASSIST	__m128i _mm_aeskeygenassist_si128 (__m128i a, constintimm8)

从以上 6 条指令中可以看出，AES-NI 可以实现密钥长度为（128 位、192 位和 256 位）的 AES 加解密，并且也支持不同模式的 AES 加解密，如 ECB、CBC 和 CTR 模式。

2.2 AES-NI 的实现效率

了解了这 6 条 AES 指令的功能与用法后，下面将测试使用 AES-NI 的 AES 实现速度，并且与未使用 AES-NI 实现的 AES 实现速度做对比。

测试环境：intel i7 - 4770 3.4GHZ 的处理器，64 位 Windows7 操作系统，编译器为 vs2010。未使用任何加速技术，编程用标准 C 语言实现。需要注意的是，在使用 AES-NI 前需要检查目标 CPU 处理上是否集成有 AES-NI，具有 AES-NI 的处理器才能做进一步的测试。C 语言使用 AES-

NI 时需要加头文件#include "wmmintrin. h"。

本节测试了使用 AES-NI 实现的 AES – 128 加密与解密的实现速度，如表 5 所示。需要注意的是，对加密与解密的测试是在已经生成 10 轮子密钥的条件下进行的。为了与普通指令实现 AES 的效率做对比，在相同平台下也测试了用普通指令实现 AES 的速度。对应的也列在了表 5 中。

表 5　AES-NI 实现 AES 加解密的效率

测试项	速度(cycles/byte)	效率比较
AES-NI 实现 AES – 128 加密	14. 6	328/14. 6≈22. 5
普通指令实现 AES – 128 加密	328	
AES-NI 实现 AES-128 解密	14. 7	331/14. 7≈22. 5
普通指令实现 AES-128 解密	331	

另外，对 AES-NI 的密钥扩展效率也进行了测试，分别测试了使用 AES-NI 生成 10 轮子密钥的耗时，AES-NI 扩展一轮子密钥的耗时，以及执行一条 AESENC 指令的耗时，即运行一轮 AES 的速度。同时，也给出了用普通指令实现它们的速度，测试数据如表 6 所示。

表 6　AES-NI 实现密钥扩展与单轮 AES 加密速度

测试项	耗时(cycles)	提升效率
AES-NI 实现 10 轮密钥扩展	1262	5322/1264≈4. 2
普通指令实现 10 轮密钥扩展	5322	
AES-NI 扩展 1 轮密钥	132	564/132≈4. 3
普通指令扩展 1 轮密钥	564	
AESENC 执行一轮 AES	15	353/15≈23
普通指令执行一轮 AES	353	

从表 5 和表 6 中的测试数据可以看出，使用 AES-NI 可以极大地提高 AES 相关算法的实现效率。在密码算法的设计中，若考虑将 AES 的轮函数或密钥扩展函数用在密码算法中，就可以极大地提高密码算法的实现效率。

3　结束语

本文介绍了 SSE2 指令集和 AES-NI 的功能与使用方式。并测试了这两种指令集的实现效率。由此发现在密码算法的设计中，若刻意使用这两种指令集，会极大地提高密码算法的实现速度。

参考文献

[1] CAESAR. Cryptographic competitions. [OL]. http://competitions. cr. yp. to/caesar. html. 2015.

[2] Wu H, Huang T. The Authenticated Cipher MORUS. [EB]. http://competitions. cr. yp. to/round1/morusv1. pdf. 2014.

[3] Wu H, Preneel B. AEGIS: A Fast Authenticated Encryption Algorithm. [EB]. http://competitions. cr. yp. to/round1/aegisv1. pdf. 2014.

[4] Intel. Intel ® 64 and IA-32 Architectures Software Developer's Manual. [EB]. http://www. intel. com/content/www/us/en/processors/architectures-software-developer-manuals. html. 2015.

[5] Intel. Intrinsics Guide. [OL]. https://software. intel. com/sites/landingpage/IntrinsicsGuide/. 2013.

[6] Intel. Intel © Advanced Encryption Standard (AES) New Instructions Set. [EB]. http://www. intel. co. jp/content/dam/doc/white-paper/advanced-encryption-standard-new-instructions-set-paper. pdf. 2010

Applications of SSE2 and AES-NI on Cryptographic Algorithm

Miao Xudong[2] **Zheng Xiulin**[1,2] **Li Yanjun**[1]

1. College of Communication Engineering, Xi'dian University, Xian Shaanxi 7100071, China

2. Beijing Electronic Science and Technology Institute, Beijing 100070, China

Abstract: The performance of the cryptographic algorithm is an important standard of measuring an algorithm. The design of traditional cryptographic algorithm is based on the bit (A5), byte (AES) or word (IDEA). Their software performance is relatively slowly. However, in the recent CAESAR competition, some cipher was designed based on fast instruction set, just like MORUS and AEGIS. Their performance was very fast. In this paper, we choose two typical instruction sets SSE2 and AES-NI. Studied the methods of two instruction set used of cryptographic algorithm, and the performance of the instruction sets. Through the study of these two kinds of instruction sets, points out the advantage of the instruction sets used in cryptographic algorithm.

Keywords: SSE2; AES-NI; instruction set

(责任编辑: 张卷美)