doi:10.11835/j.issn.1000-582X.2020.11.001

面向嵌入式系统的加密算法性能检测方法

柯亚文[a],蔡 挺[a],夏晓峰[b],向 宏[b]

(重庆大学 a.大数据与软件学院;b.信息物理社会可信服务计算教育部重点实验室,重庆 400044)

摘要:嵌入式系统信息安全是保障工业控制系统安全的必然要求,然而有限成本的硬件资源可能无法有效支撑加密算法应用所带来的额外计算开销。为研究面向嵌入式系统中加密算法移植的可行性和对系统性能的影响,提出了一个嵌入式系统加密算法性能度量方法,通过构建等效度量实现系统侧和密码侧两部分抽象层次的联系。实验利用该方法,以运行时间延迟、吞吐量和系统资源使用率为度量指标,完成对包括国产加密算法与轻量级算法等在内的多种不同配置加密算法的性能测评。实验结果表明加密算法密钥长度的增长会增加算法执行的延时,算法分组大小的增大会提高算法的运行速度,使用不同加密模式造成的性能影响随加密算法不同而表现出差异性。直接部署加密算法检测任务执行时系统的指标值可以简化测量,相比其他抽象模型在工业场景中的应用更有实际意义。

关键词:嵌入式系统;加密算法;性能检测;轻量级密码;Xilinx ZYNQ

中图分类号:TN309.7 **文献标志码**:A **文章编号**:1000-582X(2020)11-001-10

Methods of encryption algorithm performance detection oriented to embedded system

KE Yawen[a], *CAI Ting*[a], *XIA Xiaofeng*[b], *XIANG Hong*[b]

(a. School of Bigdata and Software Engineering; b. Key Laboratory of Dependable Service Computing in Cyber Physical Society, Ministry of Education, Chongqing 400044, P. R. China)

Abstract: Embedded system security is an inevitable requirement for ensuring the security of industrial control systems. However, the cost-limited hardware resources may not be able to effectively support the additional calculation overhead brought by the application of encryption algorithms. In this paper, we focusd on the feasibility of the migration of encryption algorithms for embedded systems and the impact on the system performance. And a performance benchmark method of encryption algorithm for embedded system was proposed to realize the connection between the system side and the cipher side by constructing equivalent metrics. The experiment carried out by this method covered the performance benchmark of encryption algorithms with different configurations including domestic encryption algorithm and lightweight cryptography algorithm with running time delay, throughput and system resource utilization as

收稿日期:2020-07-11

基金项目:国家重点研发计划资助项目(2017YFB0802400);"十三五"国家密码发展基金资助项目(MMJJ20180211);重庆市研究生导师团队建设项目(ydstd1821)。

Supported by the National Key Research and Development Project(2017YFB0802400), the National 13th Five Year Code Development Fund(MMJJ20180211), Chongqing Postgraduate Tutor Team Construction Project (ydstd1821).

作者简介:柯亚文(1998—),男,硕士研究生,主要从事数据安全和保密通信方向研究,(E-mail)287360811@qq.com。

通讯作者:夏晓峰(1980—),男,副教授,主要从事信息与通信安全方向研究,(E-mail)xiaxiaofeng@cqu.edu.cn。

metrics. The experimental results show that the increase in the encryption algorithm key length will increase the algorithm execution delay, and the increase in the algorithm packet size will accelerate the algorithm's running speed. The performance impact caused by the use of different encryption modes will vary with the encryption algorithm. Direct deployment of encryption algorithm to detect system index values during task execution can simplify the measurement and is more practical than the application of other abstract models in industrial scene.

Keywords: embedded systems; encryption algorithm; performance benchmark; lightweight cryptography; Xilinx ZYNQ

1 嵌入式系统信息安全

随着当代信息物理社会(CPS ,cyber physical society)的不断融合发展,信息安全越来越受到社会各级主体的重视。自上个世纪计算硬件与密码学的发展,信息安全领域已经研发出多种新颖高效且安全的加密算法,如今各类加密算法已经被广泛应用于工业控制系统(ICS,industrial control system)、航天航空、国防军事、电子商务与通讯等领域。其中,工控系统大量应用在中国关键基础设施上,关乎中国国家安全与国计民生。然而工控系统安全措施的落后,为其应对新型攻击手段与不断提高的信息安全需求带来挑战。各类嵌入式设备是工控系统的重要组成部分,承担大量运作与控制任务。嵌入式系统的信息安全关系着工控系统的整体安全。如何保证目前已有的各式加密算法或专为嵌入式设备与工控设备的新型加密算法,能有效部署移植到嵌入式系统中,在不过多影响系统固定运作的同时满足嵌入式系统的运行性能要求,并提供数据加密安全的能力,已经成为了新的挑战。

面向嵌入式系统的加密算法性能检测的研究是以工业控制系统安全作为研究的宏观背景,以工控系统中重要的组成部分–嵌入式系统为具体研究对象,研究将帮助待测试加密算法在具体嵌入式设备中的性能测试,以完善面向嵌入式系统加密算法的设计。

1.1 嵌入式系统信息安全

嵌入式系统是一种为了处理特定任务,要求具有功能专一性,计算实时性的计算机系统[1-2]。嵌入式设备同传统计算设备相比具有更加丰富的硬件结构,更注重同外接设备的互操作。

嵌入式系统上的一个主要工程问题是应付有限的资源:有限的计算处理能力、能源与内存等,这带来了工程设计优化的挑战。在CPS中,嵌入式系统通常通过物理环境影响计算的反馈环路来监视和控制物理过程,在系统中任务执行所产生的延时对于系统功能正常作业的纠正至关重要。

嵌入式系统是工控系统的重要组成部分,大量的嵌入式设备部署于工业场景,负责实际的生产工作或协调控制外接设备的运行,嵌入式系统将是安全服务的重点应用对象[3]。

1.2 嵌入式系统密码测评的意义

面向传统信息系统的加密算法研究已经有十分成熟的应用,在以保密性、完整性、可用性为优先的前提下,目前流行的加密算法不一定适合处理嵌入式系统中的任务,不能有效支撑基于有限成本的硬件运行计算需求。

研究适合有特定资源的嵌入式系统的加密算法,需要完成对目标嵌入式系统的性能测试。如准备为当前设备挑选合适的加密算法,或检测当前准备的加密算法能否部署到目标嵌入式系统,或保障加密算法应用后的嵌入式系统本身作业任务能正常进行,从而反映出对嵌入式系统进行其加密算法运行上的性能检测需求。

研究关注工控系统信息安全,面向嵌入式系统加密算法应用的巨大测评需求。从嵌入式系统测评过程中,分系统侧与密码侧两方面展开工作,根据两侧模型不同的参数配置,寻找能反映加密算法对嵌入式系统性能影响的指标,研究加密算法与嵌入式系统性能的具体依赖关系,构建加密算法性能检测的理论框架,帮助嵌入式系统选择符合运行要求的加密算法。

2　相关工作

2.1　常用加密算法介绍

对称与非对称加密算法：对称加密算法是保护信息保密性、完整性与可用性的应用最广泛的手段。对称加密解决方案可用来加密任何大小的流数据或块数据，并以明文、加密算法、对称密钥、密文与解密算法构成其五个基本组成成分。对称加密使用相同的加密密钥与解密密钥，其加密算法与解密算法的流程也大体一致。常见的对称加密算法有 DES、AES、Blowfish 与 RC5 等算法。

非对称加密所属的公钥密码学的出现带来了整个信息密码学的一次重要的革命，将原本基于置换和替换的对称加密思想转变为基于数学函数互补运算的思想。非对称加密与对称加密的明显区别在于非对称加密的加密密钥与解密密钥不相同，但加解密后又能保持明文的一致。RSA 是使用最广泛的非对称加密算法，在相同安全强度要求下，椭圆曲线密码较其他公钥密码所需的密钥规模要小得多[4-8]。与 RSA 相比，其签名速度与秘钥生成速度都更好[9]。

国产加密算法：SM 系列算法是中国自主研制的国产加密算法，又称国密算法。国密算法完整地包括了数据加密、信息摘要、身份认知、消息签名等各种基础密码算法部件。SM2 算法是基域为素域和二元扩域的椭圆曲线公钥密码算法，可满足多种密码应用中身份认证和数据完整性、真实性的安全需求[10]。SM3 杂凑算法可用作消息摘要，常应用于商业密码[11-12]。SM4 分组对称加密密码算法适用于无线局域网的安全领域[13]。

轻量级加密算法：由于微处理器和芯片体积的不断缩减，因此有必要使施加的加密方案既安全又不计算昂贵。面向资源受限设备运行的性能需求与信息安全需求，轻量级密码被认为是一种有效的加密算法设计思路。作为密码学中的子领域，轻量级密码(lightweight cryptography)主要针对的是传感器网络、嵌入式系统与 RFID 等的资源受限设备[14-16]。

2.2　常用嵌入式系统性能指标

面向嵌入式系统的加密算法性能检测涉及多项性能指标，这些指标大多与传统计算机服务器相同，但嵌入式系统作为功能专业化与资源受限的设备，其性能指标同传统运维指标相比，仅会取到运维指标的子集。根据文献分析的性能指标，以与本文嵌入式系统类别相同的操作系统性能指标为例，系统性能指标可分为以下四类[17-19]：

1)CPU 资源：包括 CPU 使用率、系统 CPU 总负载、CPU 时间软中断百分比与 CPU 每秒上下文切换次数等指标；

2)内存资源：包括物理内存可用量、缓冲区使用量、物理内存使用率与内存共享区域大小等指标；

3)读写 IO 能力：包括磁盘写入量、磁盘读取次数、磁盘 IO 占比与磁盘 IO 服务时间等指标；

4)网络通信资源：包括网络接口进出流量、TCP 连接统计数量、网络接口收发的错误包数量与 ipv4 监听状态连接数量等指标。

3　面向嵌入式系统的加密算法性能检测方法

加密算法数据处理过程复杂，信息量大，其处理过程需要占用较多的内存、CPU 使用率与能耗。加密算法应用对系统/设备侧的影响主要体现在性能、资源、可靠性与任务调度等指标上。其性能度量主要从硬件度量和软件度量考虑，分为能量消耗、延时、吞吐率 3 个维度。硬件平台的性能需求通常用等效门来表示；而软件应用的性能需求则从寄存器数目、RAM 和 ROM 的字节数等来表示。加密算法性能检测是通过密码侧和系统侧融合应用完成的，密码侧指加密算法本身处理逻辑及配置，系统侧包含嵌入式本身结构与硬件相关的性能指标。测评工作可供研究的系统测元素如图 1 所示，性能、资源、可靠性与任务调度之间互相关联，资源充足使得性能上限提高，强可靠性使得性能不会出现巨大波动以至系统崩溃，正确合理的任务调度可以使资源以及系统性能的利用最大化。密码侧对系统侧元素的影响关系如图 2 所示。

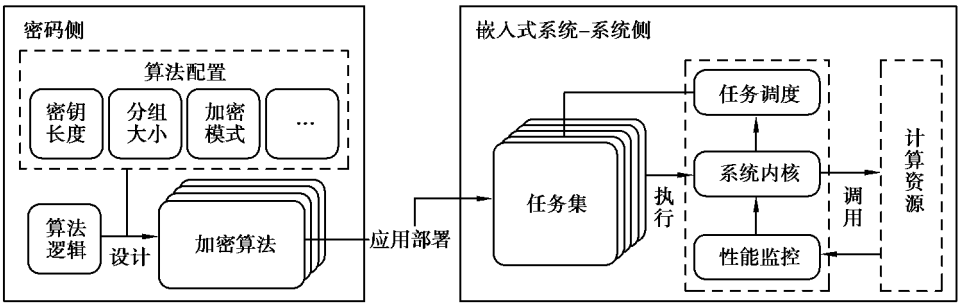

图1 系统侧与密码侧关联图

Fig. 1 System side and the cryptographic side correlation diagram

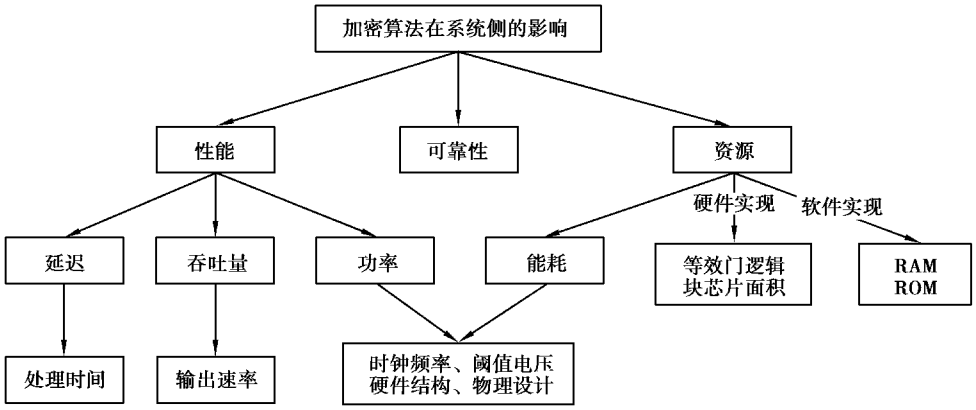

图2 密码侧对系统侧元素的影响关系

Fig. 2 The relationship between the cryptographic side and the system side elements

3.1 系统侧配置模型

对系统侧的建模存在粒度与通用性的矛盾，综合考虑几种模型之后，采用对电路或FPGA等软核进行设计的硬件超高速集成电路硬件描述语言[20](VHDL，very-high-speed Integrated circuit hardware description language)进行系统侧建模，使用时间状态机等方法来表示虚拟组件中的状态转移过程。

面向嵌入式系统的加密算法性能检测系统以工业场景为背景。工业场景中的设备在生产过程中可划分为各个层级，自顶向下可分为以西门子机床HMI为代表的监控层；控制外设运转的PLC等的控制层；实际进行生产作业的执行器或实时反馈现场数据的传感器所在的现场层，这些层级经由工业通信网络协调完成生产任务，嵌入式系统及设备存在于架构的各个层级中。

工业场景的数字孪生模型架构如图3所示，其构建过程分多步进行。首先在物理世界的单个设备实体或管理模块创建其在数字世界的虚拟仿真组件。组件中的$u(t)$与$y(t)$分别为组件间交互的输入与输出，t_0，$x(t_0)$与$x(t)$，$\dot{x}_c(t)$是虚拟组件通过数字纽带所传输的时间与状态信息数据的输入与输出，引入的时间变量t表现物理世界中，这一模块行为、状态随时间迁移而发生的变化，其中组件的状态函数$x(t)$由随时间变化的子状态序列构成。组件需定义能反映其物理性质的仿真算法Γ，例如Γ可以是示波器仿真、离散阻尼动态系统仿真、刚体动力学以及传感器仿真等。模型$M=(a, A, \Phi)$提供上级仿真算法Γ使用的参数a或组件算法A本身。模型M和仿真算法一起推演归纳出下一事件T以及可能发生的行为函数f。

选择何种状态转移模型需要考虑不同的嵌入式系统。同时，为了简化系统侧建模的复杂度，需要引入模型降阶方法，模型降阶是将高度详细和复杂的数字世界仿真模型简化，降低自由度以转移到工业场景全生命周期阶段的一项关键技术，可以在保持所需的准确性和可预测性的同时提高模型执行速度。如果仿真模型方程用下式描述

$$f(x)=T \qquad f:\mathbb{R}n \to \mathbb{R}n, x \in \mathbb{R}n, T \in \mathbb{R}n。 \tag{1}$$

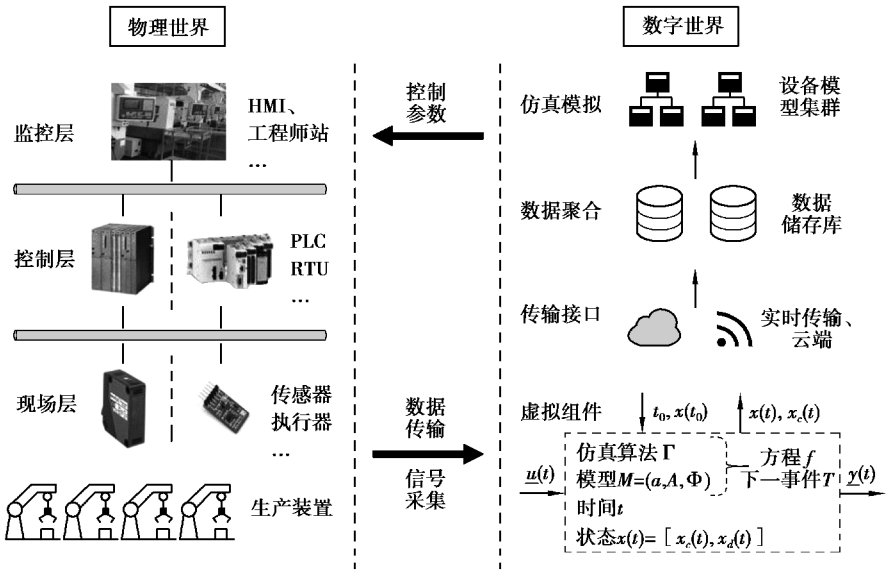

图 3　嵌入式系统组成工业场景数字孪生模型

Fig. 3　Digital twin model of industrial scene composed of embedded system

降阶后的仿真模型方程定义如下

$$f_r(x_r)=T_r, f_r:\mathbb{R}m \to \mathbb{R}m, x_r \in \mathbb{R}m, T_r \in \mathbb{R}m。 \tag{2}$$

简化模型的维度 m 应该明显小于初始模型的维度 n，模型阶数缩减必须将 $f(\cdot)$描述的初始模型方程压缩为由 $f_r(\cdot)$给出的简化方程组。对于这种简化方程组，原始状态 x 由简化状态 x_r 近似。x 和 x_r 之间的联系由投影矩阵 Q_r 给出，即

$$x_r=Q_r x, Q_r \in \mathbb{R}m \times n。 \tag{3}$$

Krylov 子空间方法是一种非常通用的降阶方法，适用于传热和结构力学等领域，可用于组件间交互的模型构建。初始状态空间的定义如下，其中 x_r，u 和 y 分别表示系统在时刻 t 的状态、输入和输出，$\boldsymbol{A}_r$，$\boldsymbol{B}_r$，$\boldsymbol{Z}_r$ 和 D 均为满足约束条件，有适当维数的常数阵。

$$E_r \dot{x_r}=\boldsymbol{A}_r x_r+\boldsymbol{B}_r u,$$

$$x_r \in \mathbb{R}m, E_r \in \mathbb{R}m \times m, \boldsymbol{A}_r \in \mathbb{R}m \times m, \boldsymbol{B}_r \in \mathbb{R}m \times p, \tag{4}$$

$$y=\boldsymbol{Z}_r x_r+\boldsymbol{D}u, \quad \boldsymbol{Z}_r \in \mathbb{R}q \times m。 \tag{5}$$

虚拟组件状态转移的信息可以使用隐马尔可夫链来创建时间序列驱动数学实体模型，该马尔可夫链通过使用一些离散状态及其转移概率来封装物理设备状态转换现象。方程 P 由状态空间 X 下的 n 个互斥间隔状态序列 U 构成，可以通过多种方式定义状态序列使 P 成立，$P(U_j)\in[0,1]$代表某一状态 U_j 的发生概率

$$P=((U_1 \cup \cdots \cup U_n=U) \wedge (U_j<U_{j+1} \mid \forall j \in \{1,\cdots,n-1\}) \wedge,$$
$$(U_k \cap U_l=\mid \forall k \in \{1,\cdots,n\}, \forall l \in \{1,\cdots,n\}-\{j\}))。 \tag{6}$$

虚拟组件依赖于传输接口适配数字纽带完成数据流通与信号采集，数字孪生系统对数据传输具有强实时性要求，这可能会为嵌入式系统引入额外的网络通信开销。系统侧建模以虚拟组件为基础，结合其他时间序列或状态机，可满足通用性与有效性的建模需求。

3.2　密码侧算法配置模型

密码侧指加密算法的处理逻辑及配置，本部分基于对称加密算法进行分析。常见的配置要求集中在密钥长度与加密模式上，密码侧的密钥长度与可使用的分组长度密切相关，一般出于单次分组加密的安全性与成本开销的考虑，每个算法都固定设计了分组大小，因此对大部分密码侧算法来说，更改其分组大小配置会直接影响算法本身架构。

在分组大小固定的情况下，考虑单次和分组大小相同大小的数据块的密码侧建模。首先将密码侧加密

算法本身全部处理逻辑定义为一项规则逻辑映射 $E(\cdot)$，每一轮的处理过程为 $g(\cdot)$，每一轮之间的操作关系为 $p(\cdot)$，加密轮数表示为 N，初始分组大小明文向量为 $\boldsymbol{x}$，转变为输入状态矩阵为 $\boldsymbol{X}$；密钥字节向量为 $\boldsymbol{k}$，扩展密钥矩阵为 $\boldsymbol{K}$，提供给每一轮的密钥矩阵为 $\boldsymbol{K}_i$，每一轮的变换操作可由矩阵表示为 $\boldsymbol{A}$。则单分组密码侧加密算法总体流程表示为

$$E(\boldsymbol{X},\boldsymbol{K})=p(g(\boldsymbol{A},\boldsymbol{X}_i,\boldsymbol{K}_i),N),i\in 1\cdots N\ 。\tag{7}$$

引入加密模式到单分组密码侧加密算法流程中，定义加密模式过程为 $T(\cdot)$，最终密文为 C，则密码侧加密算法模型可表示为

$$C=T(E(\boldsymbol{X}_i,\boldsymbol{K})),\quad i\in 1\cdots N。\tag{8}$$

不同的密码侧配置会为密码侧模型执行带来不同影响。每个加密算法的轮处理过程是不相同的，因此带来了 $g(\cdot)$复杂性上的区别。密码侧配置与算法处理逻辑 $g(\cdot)$都会对系统侧的性能产生影响，安全性越高的算法，$g(\cdot)$的处理逻辑一般越复杂，或者分组大小越大，因为较大的分组可以使单个分组的数据离散到更大的数据空间。

密钥长度与分组长度强相关，往往是以分组长度来确定所需的密钥长度。当分组长度一定时，密钥长度对密码侧的影响体现在：本身生成初始密钥的时间和内存要求增加、密钥扩展时间和内存的增加与加密轮数的增长。密钥扩展变化和加密轮数变化相关，首先带来的是安全强度的改变，其次是线性时间与内存上的由于密钥扩展增加所带来额外循环次数的增大。密钥长度与分组长度为密码执行侧带来的线性变化可由较大的明文内容的输入来直观发现。

密码侧模型同样需要考虑随机数的选取方式，包括 IV(初始向量)值与 salt(盐值)的确定。这方面密码侧配置的影响也同初始化的复杂度相关，但对密码侧应用的时间或内存的影响同整体应用开销考虑仅占极小一部分。

采用不同的加密模式会改变 T(·)的复杂程度。最简单的加密模式是电话本 ECB 模式，分组之间独立加密，因此不会产生其他额外的开销，密文分组链接 CBC 与密文反馈 CFB 模式分别对应分组与数据流，CFB 需要考虑额外的伪随机数输出同明文异或生成下一单元密文的开销。输出反馈 OFB 模式与 CFB 类似，但是将加密算法的输入变为是上一次加密的输出，且采用的是分组而不是流密码。最后比较特别的是计数器 CTR 模式，密码侧需要引入计数器，并增加对计数器的操作。

3.3 嵌入式系统加密算法性能指标度量

加密算法在嵌入式系统中占用大量计算资源与通信资源，在高级抽象层次进行嵌入式系统架构建模与加密算法建模，在密码侧模型与系统侧模型之间的注入语义对等等效度量，其逻辑结构如图 4 所示，选用典型的嵌入式设备进行试验验证，进而帮助嵌入式设备挑选合适的加密算法或优化新设计的面向嵌入式系统的加密算法。

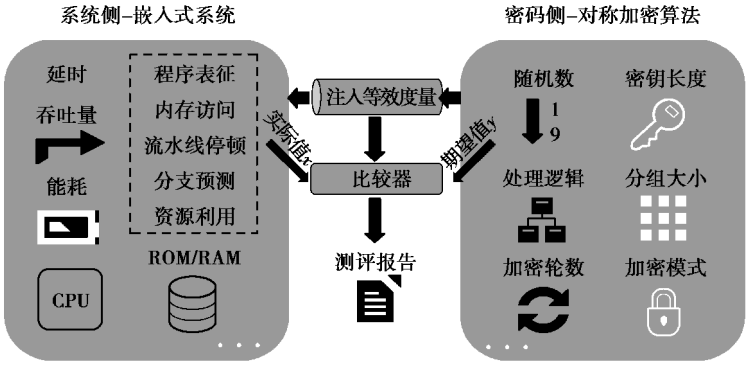

图 4 系统侧与密码侧的高层次抽象框架

Fig. 4 High-level abstract frame on the system side and the cryptographic side

系统侧性能度量模型中，性能事件可以分为五类：程序表征，内存访问，流水线停顿，分支预测和资源利用[21]。程序特征描述事件有助于定义程序的属性，这些属性在很大程度上独立于处理器的实现。这些事件的最常见示例是程序完成的指令的数量和类型。内存访问事件通常包括处理器内存层次结构的最大事件类

别和辅助性能分析。管道停顿事件信息可帮助用户分析程序指令在管道中的流动情况。具有深度流水线的处理器严重依赖于分支预测硬件来使管道充满有用的指令。分支预测事件允许用户分析分支预测硬件的性能，例如通过提供错误预测分支的计数。资源利用率事件允许用户监视处理器使用某些资源的频率。

系统侧能耗的计算，可以由式(9)来计算，其中结合了数字孪生建模中的状态序列转移概率。式中 P_i 和 P_{ij} 分别表示每个状态和每个转换的功耗，并且 π_i 是稳态概率，而 λ_{ij} 是与状态 i 和 j 之间的转换相关联的转换速率。

$$P(k)=\sum\nolimits_{alli}\pi_i \cdot P_i+\sum\nolimits_{alli,j}\lambda_{ij} \cdot P_{ij}。\tag{9}$$

加密算法对嵌入式设备性能或调度的影响，从系统侧与密码侧的属性并集中发现的中间变量的等效度量为运行时间或延时。为提供密码服务而应用的加密算法，其执行时间将造成运行时间的改变，进一步影响任务调度或性能。在密码侧对运行时间改变的度量需要考虑嵌入式本身硬件与系统模型，可以直接部署加密算法检测任务执行前后的时间间隔，得到运行时间的变化或者加上多次实验中时间变化的抖动。同时，选用执行时间或延时作为等效度量能满足简化测量方式的需求，因此实验的核心部分是测量密码侧部署于系统侧后的执行时间。

测试了嵌入式系统进程级与系统级的性能指标，进程级的性能指标包括执行时间、吞吐量、CPU 使用率、内存使用量与内存使用率，系统级的性能指标包括 CPU 利用率、内存利用率、网络传输字节数与网络接收字节数。

进程级性能指标使用 Petalinux 自带的 time 命令获得，当无其他参数指示时，time 命令执行后续进程，返回命令执行的花费时间(real time)，其值不等于内核态时间与用户态时间的简单相加，此时根据得到的运行时间，以及加密对象明文的大小，可以计算该加密算法的吞吐量

$$\text{throughput(Mbytes/}s\text{)}=\text{File Size(bytes)}\div\text{real time(ms)}\div 1\,000。\tag{10}$$

CPU 使用率与内存信息可以使用带－v 参数的 time 命令得到，提供了该进程获得 CPU 时间的占比与运行时最大内存占用量。根据嵌入式系统可提供的最大物理内存大小，结合定义可计算得到内存使用率

$$\text{mem used pct}=\text{mem used}\div\text{max physical memory size}\times 100\%。\tag{11}$$

系统级性能指标的获取通过系统提供的相关文件信息完成。Petalinux 可通过读写/proc/stat 文件获取 CPU 信息，读写/proc/meminfo 获取内存信息，读取/proc/net/dev，指定网络适配器获取网络信息。CPU 利用率的合理范围较大，85%的利用率可以作为一个正常阈值，当内核态或用户态占据过多的 CPU 时间，会达到 CPU 性能瓶颈，导致系统的响应时间大幅上升。

通过读取/proc/stat 的信息计算 CPU 利用率，需要在很短的时间内 2 次测量读取 cpu 的空闲时间和总处理时间，并结合定义计算 CPU 利用率

$$\text{CPU used pct}=\left(1-\frac{\text{idle2}-\text{idle1}}{\text{total2}-\text{total1}}\right)\times 100\%。\tag{12}$$

4　实验与结果

将对面向嵌入式系统的加密算法性能检测的结果做具体分析，提供的多种不同配置的加密算法列表如下图所示，实验提供了对传统加密算法、国密算法 SM2 与 SM4、以及轻量级分组加密算法的面向嵌入式系统的性能测评。目前已实现的支持测评的加密算法及其配置信息如表 1 所示，测试共有 188 种组合。

表 1　测评支持的加密算法与配置表

Table1　Encryption algorithm and configuration table supported by benchmark

加密算法	密钥长度(位)	加密模式	分组大小(位)	其他
AES	128/192/256	CBC/CFB/CTR/ECB/OFB	128	10/12/14 轮数
Blowfish	128	CBC/CFB/ECB/OFB	64	16 轮数
Camellia	128/192/256	CBC/CFB/ECB/OFB	128	

续表1

加密算法	密钥长度(位)	加密模式	分组大小(位)	其他
CAST5	128	CBC/CFB/ECB/OFB	64	CAST-CBC
DES	64	CBC/CFB/ECB/OFB	64	DES3
DES-EDE	192	CBC/CFB/ECB/OFB	64	
DES-EDE3	192	CBC/CFB/ECB/OFB	64	
DESX	—	CBC/ECB	—	
RC2	40/64	CBC/CFB/ECB/OFB	64	
RC4	40	ECB	64	
Seed	—	CBC/CFB/ECB/OFB	—	
SM2	256	—	—	
SM4	128	CBC/EBC	128	
Simon	64/72/96/128/144/192/256	CBC/PCBC/CFB/CTR/ECB/OFB	32/48/64/96/128	
Speck	64/72/96/128/144/192/256	CBC/PCBC/CFB/CTR/ECB/OFB	32/48/64/96/128	

以AES加密算法为例子做具体分析,AES加密算法将在密钥长度、加密轮数与加密模式等具有不同的配置,在嵌入式系统应用影响的性能指标将考虑延时(处理时间)、吞吐量、CPU使用率与内存使用量等,待加密明文大小为8.31 MB。同一算法配置的不同,将显著地改变运行时性能。影响最大的是加密算法应用所带来的延时,密钥长度与加密轮数的增加将导致嵌入式系统需要更多的时间去处理增加的工作量。选用不同的加密模式会得到不同的效果,使用简单的ECB电话本模式,会得到较短的处理延时与最少的内存使用量,而使用CTR计数器模式可以得到最高的吞吐量与最短的延时。但无论以何种配置运行AES算法,目标嵌入式系统的CPU使用率都未出现较大波动,可能的原因在于CPU使用率受限于当前存在的进程数,实验数据生成过程中,只有加密算法应用检测,与模拟代表工业场景数据采集发送的系统级性能检测程序这两个进程运行,在Petalinux嵌入式系统资源较为充足的环境下分占CPU时间。不同配置的AES算法应用的性能测评结果如下表2所示。

表2 不同配置AES算法性能检测结果

Table 2 AES algorithm performance test results of different configurations

密钥长度	加密轮数	加密模式	延时/ms(标准差)	吞吐量(MB/s)	CPU/%	内存占用/KB
128位	10轮	CBC	511.48(3.67)	17.044	50.58	12 285.44
128位	10轮	CFB	510.12(3.73)	17.089 5	49.78	12 434.88
128位	10轮	CTR	434.3(3.84)	20.073 5	49.87	12 340.16
128位	10轮	ECB	488.36(3.40)	17.850 9	49.98	12 082.24
128位	10轮	OFB	504.6(3.17)	17.276 2	50.49	12 231.68
192位	12轮	CBC	569.32(3.93)	15.312 4	49.98	12 267.84
192位	12轮	CFB	571.2(4.31)	15.262 1	50.20	12 324.16
192位	12轮	CTR	487.54(4.19)	17.881 4	49.92	12 227.52

续表 2

密钥长度	加密轮数	加密模式	延时/ms(标准差)	吞吐量(MB/s)	CPU/%	内存占用/KB
192 位	12 轮	ECB	558.84(3.86)	15.599 6	50.15	12 179.84
192 位	12 轮	OFB	565.52(3.31)	15.415 1	50.73	12 299.84
256 位	14 轮	CBC	631.66(4.02)	13.801 1	50.52	12 342.08
256 位	14 轮	CFB	632.48(3.83)	13.783 1	49.85	12 288.32
256 位	14 轮	CTR	540.98(3.63)	16.114 5	50.17	12 230.08
256 位	14 轮	ECB	620.24(3.14)	14.055	49.98	12 075.52
256 位	14 轮	OFB	625.12(3.44)	13.945 3	49.91	12 148.8

图 5 展示了不同加密模式与密钥长度的 AES 算法在嵌入式系统中的延时，CBC、CFB 与 OFB 加密模式在延时上的结果大致相同，相比较之下 ECB 模式较低，而 CTR 计数器模式达到了最好的延时性能。同时，由图可知在 AES 算法中，密钥长度的增长会带来延时上的额外开销。

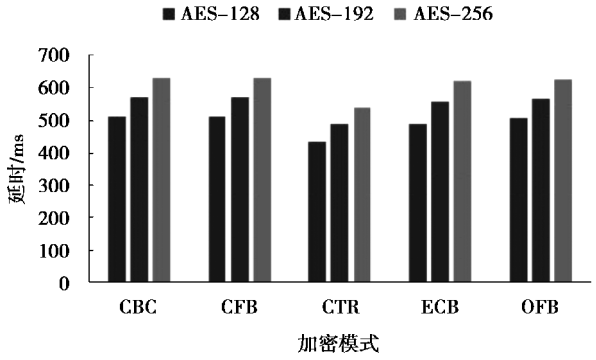

图 5　不同加密模式与密钥长度的 AES 算法执行延时

Fig. 5　AES algorithm execution delay for different encryption modes and key lengths

实验结果表明，密钥长度的增加会导致延时的增大，分组大小的增加可以加快加密算法的运行速度，密钥长度、分组大小及加密模式的影响对不同的加密算法表现出差异性。嵌入式系统的 CPU 使用率与内存占用受加密算法本身配置的影响有限，取决于实际的加密任务场景以及明文数据块大小。

5　结　论

提出了一个面向嵌入式系统的基于 Xilinx ZYNQ 的加密算法性能检测系统框架与方法。ZYNQ 嵌入式架构集成了 ARM 处理器与 FPGA 技术，已普遍应用于真实工业场景中。从系统侧与密码侧 2 个模型介绍在嵌入式系统加密算法应用性能上的研究。密码侧模型考虑加密算法本身处理逻辑与算法配置，算法配置包括密钥长度、分组大小、加密模式、加密轮数与随机数生成等，系统侧的性能指标主要关注处理时间、吞吐量、CPU 占用率、能耗与内存使用量等。最后提出了一种嵌入式系统加密算法性能指标度量的框架，以延时作为系统侧密码侧的等效度量，完成对加密算法应用性能影响的分析。

参考文献：

[1] Michael Barr. Embedded systems glossary[M]. Ethiopia: Neutrino Technical Library, 2007.

[2] Heath, Steve. Embedded systems design[M]. Netherlands: Newnes, 2003.

[3] Langner R. Stuxnet: dissecting a cyberwarfare weapon[J]. IEEE Security & Privacy Magazine, 2011, 9(3): 49-51.

[4] Vračar L M, Stojanović M D, Stanimirović A S, et al. Influence of encryption algorithms on power consumption in energy

harvesting systems[J]. Journal of Sensors, 2019, 2019: 1-9.

[5] Nandi A, Marcuelescu R. System-level power/performance analysis for embedded systems design[C]// DAC'01: Proceedings of the 38th annual Design Automation Conference. New York, USA: ACM Press, 2001: 599-604.

[6] Mohanty S, Prasanna V K. Rapid system-level performance evaluation and optimization for application mapping onto SoC architectures[C]// 15th Annual IEEE International ASIC/SOC Conference. Piscataway, NJ: IEEE, 2002:160-167.

[7] Zelenova S A, Zelenov S V. Schedulability analysis for strictly periodic tasks in RTOS[J]. Programming & Computer Software, 2018, 44(3):159-169.

[8] Lee E, Seshia S. Introduction to embedded systems-a cyber-physical systems approach[M]. Cambridge, MA: Mit Press, 2016.

[9] Douglas R S. 密码学原理与实践[M]. 冯登国,译. 北京: 电子工业出版社, 2003: 131-142.
Douglas R S. Cryptography theory and practice[M]. FENG Dengguo trans. Beijing: Publishing House of Electronics Industry, 2003: 131-142. (in Chinese)

[10] 冯登国. 国内外密码学研究现状及发展趋势[J]. 通信学报, 2002, 23(5): 18-26.
FENG Dengguo. Status quo and trend of cryptography[J]. Journal on Communications, 2002, 23(5): 18-26. (in Chinese)

[11] 赵军, 曾学文, 郭志川. 支持国产密码算法的高速 PCIe 密码卡的设计与实现[J]. 电子与信息学报, 2019, 41(10): 2402-2408.
ZHAO Jun, ZENG Xuewen, GUO Zhichuan. Design and implementation of high speed PCIe cipher card supporting GM algorithms[J]. Journal of Electronics & Information Technology, 2019, 41(10): 2402-2408. (in Chinese)

[12] GM/T 0004-2012, SM3 密码杂凑算法[S]. 北京: 中国标准出版社, 2012.
GM/T 0004-2012, SM3 cryptographic hash algorithm[S]. Beijing: China Standard Press, 2012. (in Chinese)

[13] GM/T 0002-2012, SM4 分组密码算法[S]. 北京: 中国标准出版社, 2012.
GM/T 0002-2012, SM4 block cipher algorithm[S]. Beijing: China Standard Press, 2012. (in Chinese)

[14] Bafandehkar M, Yasin S M, Mahmod R, et al. Comparison of ECC and RSA algorithm in resource constrained devices[C]// 2013 International Conference on IT Convergence and Security (ICITCS). Piscataway, NJ: IEEE, 2013: 1-3.

[15] Weidler N R, Brown D, Mitchell S A, et al. Return-oriented programming on a resource constrained device[J]. Sustainable Computing: Informatics and Systems, 2019, 22: 244-256.

[16] McKay K A, Bassham L, Turan M S, et al. Report on lightweight cryptography[R]. New York, USA: National Institute of Standards and Technology, 2017.

[17] Jararweh Y, Tawalbeh L, Tawalbeh H, et al. Hardware performance evaluation of SHA-3 candidate algorithms[J]. Journal of Information Security, 2012, 3(2): 69-76.

[18] Rhett S. Cryptography concepts and effects on control system communications[C]// Sensible Cybersecurity for Power Systems: A Collection of Technical Papers Representing Modern Solutions, 2018.

[19] Vračar L M, Stojanović M D, Stanimirović A S, et al. Influence of encryption algorithms on power consumption in energy harvesting systems[J]. Journal of Sensors, 2019, 2019: 1-9.

[20] Chu P P. FPGA prototyping by VHDL examples: Xilinx microBlaze MCS SoC[M]. Hoboken, USA: John Wiley & Sons, 2017.

[21] Sprunt B. The basics of performance-monitoring hardware[J]. IEEE Micro, 2002, 22(4): 64-71.

(编辑 侯 湘)