Application Note

在 Sitara MCU 器件上集成 MbedTLS

Shaunak Deshpande

摘要

TI Sitara™ MCU 器件具有强大的网络协议栈和已在业界广泛使用的硬件 IP 支持。如果无法保证网络的安全性，那么与外部世界的连接就会存在风险。虽然网络功能和硬件 IP 正在持续不断地发展，逐步变得更加高效和优化，但安全方面也不容忽视。缺乏安全性可能会导致系统功能异常，甚至导致环境易于受到 MITM、窃听、篡改或消息伪造等网络攻击。这种不足可以通过使用传输层安全协议 (TLS) 来弥补。TLS 是一种加密协议，主要通过对传输的数据进行加密来为互联网通信提供保护。TLS 确保可在网络中的两个或多个实体之间建立安全的通信通道。

TLS 协议可分为两个部分：

- 握手层：该层负责执行 TLS 握手，并在实体经过验证且握手完成后更改密码规范。
- 记录层：该层负责对要传输的数据进行分段、压缩、身份验证和加密。

本文档介绍了在现有 LwIP TCP/IP 网络协议栈上集成适用于 Sitara MCU 器件的 MbedTLS，从而增加传输层（OSI 模型的 L4）的安全性。传输层提供了一个安全的端到端通信通道。因此，所有数据都可以在传输层（第 4 层）之后通过网络安全地传输。MbedTLS 项目作为独立库移植到 TI 架构中，可测量性能以及通过硬件加速优化加密操作的方法。此外，本文档还讨论了 MbedTLS 在网络安全示例中的使用。

本文档中讨论的代码和示例可以在适用于 AM243x、AM263x、AM273x 和 AM64x 器件的 TI MCU_PLUS_SDK v09.00 或更高版本中找到。

内容

插图清单

表格清单

商标

1 引言

工业 4.0 属于数据密集型应用并依赖于实时决策，相关应用需要一种机制来安全地通过网络传输数据。该演示是 TI Sitara MCU 器件在网络和安全领域的一项进步。

本应用手册介绍了以下内容：

- 向用户介绍 MbedTLS
- 演示 MbedTLS 如何与 LwIP 配合使用
- 举例说明基于 MbedTLS 构建的网络安全

TI Sitara MCU 器件具有高性能多核处理能力，专为实时处理和连接而设计。MbedTLS 与 LwIP 的集成旨在通过增加一层安全性来进一步增强网络和连接性。LwIP 是一种嵌入式系统中常用的轻量级 TCP/IP 堆栈。这些器件具有用于联网的 CPSW 和 ICSS IP。这些器件具有原始处理能力和实时信号控制能力，并支持多协议以太网标准，因此网络的运行速度可高达 1Gbps。这些器件还具有强大的加密加速器，可将加密工作从软件卸载到硬件，从而进一步优化整体应用的性能。

1.1 本文档中使用的首字母缩写词

1. **AES**：高级加密标准 (Advanced Encryption Standard)
2. **CA**：证书颁发机构 (Certificate Authority)
3. **CPSW**：通用平台以太网交换机 (Common Platform Ethernet Switch)
4. **DER**：区分编码规则 (Distinguished Encoding Rules)
5. **DES**：数据加密标准 (Data Encryption Standard)
6. **ECDSA**：椭圆曲线数字签名算法 (Elliptic Curve Digital Signature Algorithm)
7. **FOTA**：无线固件升级 (Firmware Over the Air)
8. **HTTP**：超文本传输协议 (Hyper Text Transfer Protocol)
9. **ICSS**：工业通信子系统 (Industrial Communication Subsytem)
10. **MITM：**中间人攻击 (Man In The Middle attack)
11. **MQTT**：消息队列遥测传输 (Message Queuing Telemetry Transport)
12. **OSI：**开放系统互连 (Open System Interconnection)
13. **PBUF**：数据包缓冲区 (Packet Buffer)
14. **PCB**：协议控制块 (Protocol Contol Block)
15. **PEM**：隐私增强邮件 (Privacy Enhanced Mail)
16. **RSA**：Rivest-Shamir-Adleman
17. **SHA**：安全哈希算法 (Secure Hash Algorithm)
18. **SSL**：安全套接字层 (Secure Socket Layer)
19. **TCP：**传输控制协议 (Transmission Control Protocol)
20. **TLS：**传输层安全协议 (Transport Layer Security)

2 MbedTLS

2.1 MbedTLS 是什么？

MbedTLS 是 SSL 和 TLS 协议的实现，以及必要的支持代码和相应的加密算法。MbedTLS 根据 Apache 许可证 2.0 版进行分发。MbedTLS 提供了通过 TCP/IP 协议栈进行安全通信所需的抽象层，如图 2-1 所示。MbedTLS 和 LwIP 网络安全示例所需的软件栈。

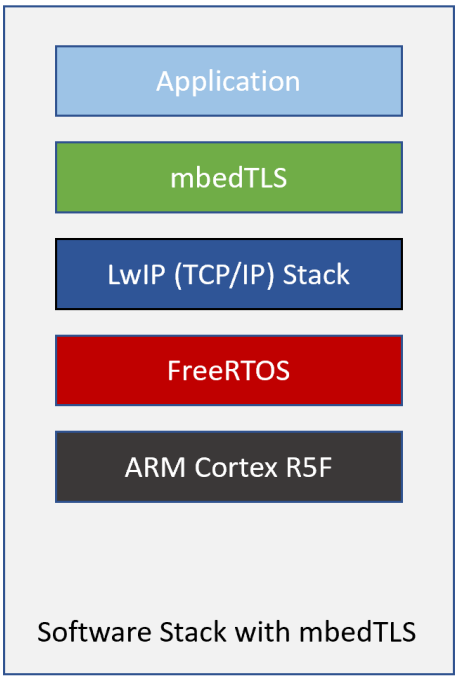

图 2-1. 基于 TCP/IP 协议栈的 MbedTLS 安全通信

2.2 为何选择 MbedTLS？

MbedTLS 的替代品包括 OpenSSL、GnuTLS、LibreSSL，但 TI 更喜欢使用 MbedTLS，因为 LwIP 中提供对 MbedTLS 的直接支持。只需对现有 LwIP 配置进行极少的更改，即可将 MbedTLS 移植到 LwIP 中。

MbedTLS 构建为一个独立的库，在 LwIP 中具有用于应用层 API 的挂钩，下一节将对此进行详细讨论。MbedTLS 提供详细的文档和易于使用的 API，是一种快速高效地为应用增添安全性的方法。MbedTLS API 不仅可用于提高网络通信的安全性，还可用于需要加密实现的应用。

此外，MbedTLS 提供了一种定义明确的方法来将加密操作从软件卸载到硬件。请注意，AES、SHA、RSA、ECDSA、ENTROPY、计时函数、DES、ARIA 等均可以卸载到硬件上。MbedTLS 还提供了部分卸载加密工作的选项，例如将加密卸载到硬件而在软件上执行解密，从而提高了优化的整体灵活性。根据 CPU 时钟速度、RAM 执行、缓存使用情况等因素，可以优化吞吐量。对 TI Sitara™ AM2434 进行相同测试后发现，与执行软件加密相比，使用硬件加密加速器进行 AES 加密和解密时吞吐量增加了 8 至 10 倍（注意：这不是硬件卸载加密性能的基准数据，这些性能数据特定于实现并可以进一步改进）。MbedTLS 配置由头文件控制，该文件可用于启用或禁用所需的加密模块甚至密码套件。这可用于减小 MbedTLS 库的代码大小。

2.3 MbedTLS 的应用

借助 MbedTLS，我们为 TI 器件的开放世界网络增添了安全性。MbedTLS 可用于网络中任何需要填补 TLS 空白的场合。MbedTLS 可用于将设备用作 TLS 实体（客户端或服务器）。从应用的角度来看，MbedTLS 功能可用于固件无线 (FOTA) 更新。其他潜在的应用包括远程诊断、车辆状况监控、安全外壳和安全日志、安全信息以及事件管理等。MbedTLS 是 OpenVPN 集成中的一个潜在用例。

3 在 Lwip 上使用 MbedTLS

轻量级 TCP/IP (LwIP) 协议栈是嵌入式系统中使用的开源 TCP IP 协议栈。LwIP 协议栈旨在减少内存用量和缩小代码大小。LwIP 采用分层方法。针对层间通信的宽松方案是通过使用共享内存建立的，这意味着应用进程和网络代码可以使用相同的内存或内部缓冲区。内部数据包在 LwIP 中表示为 Pbufs（数据包缓冲区）。这些 Pbuf 包含要在网络中传输的数据包的有效负载。在 MbedTLS 不存在的情况下，Pbuf 包含要传输的实际应用数据。但在安全通信的情况下，需要通过加密、哈希处理等多种可能方法之一来保护这些数据。

LwIP 工程已经内置了对 MbedTLS 的支持。启用 MbedTLS 后，通过默认 TCP 层传输的数据现在采用备用路由并由备用 TCP/IP 层（也称为 ALTCP 层）进行处理。这些层设置了回调函数。回调函数决定应用在建立连接、发送

数据、接收数据、关闭连接、处理错误等方面的行为。MbedTLS 可与 LwIP 套接字 API、BSD 套接字 API 和 Netconn API 一起使用。这种广泛的用途是通过填充通用 altcp_tls_config 结构来实现的，该结构在内部使用 MbedTLS 函数进行加密/SSL 操作。

altcp_tls_config 结构保存创建新 TLS 客户端或服务器连接所需的状态。然后，此 TLS 配置会直接传递给 LwIP 函数或其他 LwIP 内部结构。例如，通常通过回调 tcp_recv（未启用 MbedTLS 时）传递的数据现在会通过 altcp_recv（启用了 mbedTLS 时）传递。这些替代实现包含额外的代码（默认在 lwip-stack 内）。所有替代层实现都在以下路径的文件中定义：*MCU_PLUS_SDK/source/networking/lwip/lwip-stack/src/apps/altcp_tls*。

MbedTLS 与 LwIP 的用法由 lwip-config/lwipopts.h 中定义的宏在顶层进行控制。在此启用 MbedTLS 即会启用 ALTCP 层。当 LwIP 库重建（MbedTLS 已启用）时，ALTCP 层将处理数据。如图 3-1 所示，启用了 MbedTLS 时，数据通过备用 TCP 层 (altcp) 传输。启用了 MbedTLS 时，数据通过备用 TCP 层 (altcp) 传输。

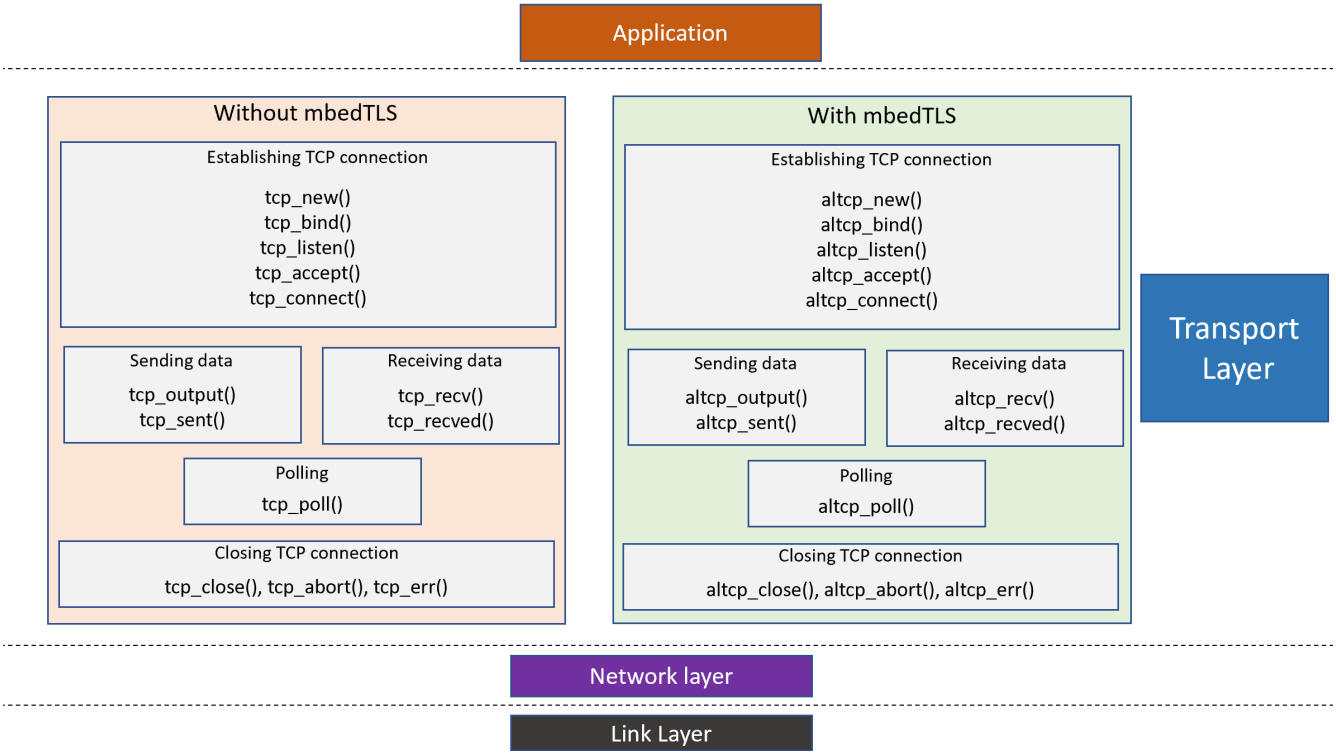

图 3-1. 与 LwIP 集成时的 MbedTLS 功能

当调用 tcp_new() 函数时，会创建一个新的协议控制块 (TCP PCB)。此 PCB 是一个新的 TCP 连接标识符，可以设置用于侦听新的连接或显式连接到另一个主机。在使用 MbedTLS 的情况下，这时会使用备用 PCB（即 altcp_pcb，而不是 tcp_pcb）。altcp_pcb 结构包含 altcp_functions、对下一个 PCB 的引用、指向参数的指针、PCB 的状态以及应用程序回调。

LwIP 提供 HTTP 客户端和服务器、MQTT 客户端等示例应用。普通的 LwIP 应用不安全，网络中的通信也不安全，这在现场会导致设备容易成为网络攻击的目标。

在 LwIP 上使用 MbedTLS 可确保数据通过安全的端到端传输通道传递。MbedTLS 会在传输实际数据之前执行 3 路 TLS 握手。TLS 握手可确保愿意通信的实体经过授权且可信。TLS 可确保数据的加密和完整性，以及网络中实体的身份验证。此外，MbedTLS 还可在需要额外安全性时提供双向 SSL 身份验证。

启用 MbedTLS 后，LwIP 的内存占用量变化几乎可以忽略不计。据观察，应用中 MbedTLS 本身的内存占用量约为 176KB（还可以进一步降低）。表 3-1 展示了内存占用量。

表 3-1. MbedTLS 库内存占用量

代码	RO 数据	RW 数据	总大小（字节）
127988	39289	9159	176436

3.1 TLS 服务器示例（HTTPS 服务器）

该示例演示了 AM2x 器件用作 HTTPS 服务器的情形，该服务器接受客户端连接并发回固定响应。在此示例中，MbedTLS 与 LwIP 搭配使用，来实现完整的功能。

下面介绍了 HTTPS 服务器的顶层工作方式：

MbedTLS 的作用：

- 创建使用服务器证书、私钥、私钥访问密码（可选）的 TLS 配置。标准分配器函数为 TLS over TCP 创建 altcp PCB。
- 加载证书和私钥。然后基于 x509 证书（DER 或 PEM）的格式解析证书和密钥。
- 将私钥的模数与证书中公钥的模数进行比较。使用可信 CA 的私钥对证书进行签名。
- 使用公钥来验证上述签名。如果验证完成，则执行后续步骤。
- 将 TLS 配置传递给 LwIP 应用 API，这些 API 在 PCB（相应 TCP 连接的进程控制块）内部使用相同的 TLS 配置。
- 使用 mbedTLS 加密函数验证证书和密钥。如果发生任何解析错误或数据不一致，则表示证书和密钥无效，正在进行的网络连接将被断开。

LwIP 的作用：

- 设置 LwIP PCB，然后将该 PCB 绑定到定义的端口（采用 HTTPS 时为 8080）。然后，PCB 将连接绑定到本地端口号和 IP 地址，本例中由 DHCP 服务器获得这些信息。
- 然后，将 TCP 连接的状态设置为 LISTEN（用于接受新连接的模式）。
- 设置 altcp_accept 回调，用于处理新的传入连接、分配内存来管理连接状态，以及设置用于发送、接收、错误处理和轮询的回调。
- http_recv 回调在获得数据时通知 TCP PCB 已接收到数，然后解析数据。
- http_poll 回调每 2 秒轮询一次连接的另一端，如果 8 秒内没有接收到数据，连接就会关闭。
- 在出现错误时，http_err 回调关闭连接并释放资源。
- http_sent 回调函数负责发送数据并从远程主机获取确认。

图 3-2 展示了使用 SA2UL 加密加速器进行加密（可选）时的相同用例。另一种选择是使用 MbedTLS 软件加密技术，而不是将其卸载到硬件上。

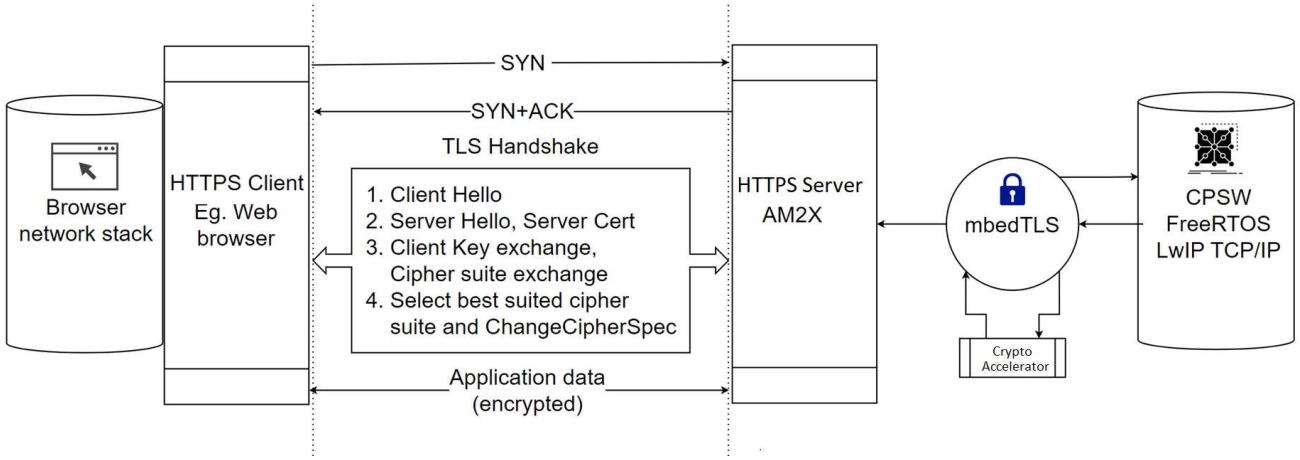

图 3-2. HTTPS 服务器示例的端到端工作方式概述

有关更多详细信息，请参阅：AM243x MCU+ SDK： CPSW Lwip HTTPS 服务器示例

3.2 TLS 客户端示例（MQTT 客户端）

该示例演示了 AM2x 器件用作 MQTT 客户端的情形，其中客户端订阅 MQTT 代理并在另一个客户端发布数据时截取数据。本例中使用的 MQTT 代理是开源软件 Mosquito MQTT。分配给 Mosquito 代理和 MQTT 客户端的 IP 地址是静态地址。

此示例中执行双向身份验证，其中涉及到客户端验证服务器和服务器验证客户端。由于使用本地生成的证书和自签名证书，因此 CA 证书和 CA 密钥用作服务器证书和密钥。

下面介绍了 MQTT + TLS 客户端的顶层工作方式：

MbedTLS 的作用：

- 创建使用 CA 证书、CA 密钥、客户端证书、私钥、私钥访问密码（可选）的 TLS 配置。标准分配器函数为 TLS over TCP 创建 altcp PCB。
- 加载客户端证书和私钥。然后基于 x509 证书（DER 或 PEM）的格式解析证书和密钥。
- 传递私钥并与证书中的公钥进行比较。如果两者匹配，则证书和密钥/密钥对验证成功。
- 将 TLS 配置传递给 LwIP 应用 API，这些 API 在 PCB（相应 TCP 连接的进程控制块）内部使用相同的 TLS 配置。
- 使用 mbedTLS 加密函数验证证书和密钥。如果发生任何解析错误或数据不一致，则表示证书和密钥无效，正在进行的网络连接将被断开。

LwIP 的作用：

- 在 LwIP 的 mqtt_context 结构内填充 TLS 配置。
- 为定义的 MQTT 端口和 IP 地址创建 altcp PCB，并设置连接回调。传递包含 TLS 配置的客户端信息。
- 设置 altcp_accept 回调，用于处理新的传入连接、分配内存来管理连接状态，以及设置用于发送、接收、错误处理和轮询的回调。
- mqtt 连接回调负责订阅/取消订阅一个主题，并将有关该主题的所需数据发送给 MQTT 代理。
- 设置输入回调函数，用于确定客户端在接收到数据时的行为。
- 根据连接状态处理数据。

图 3-3 展示了使用 SA2UL 加密加速器进行加密（可选）时的相同用例。

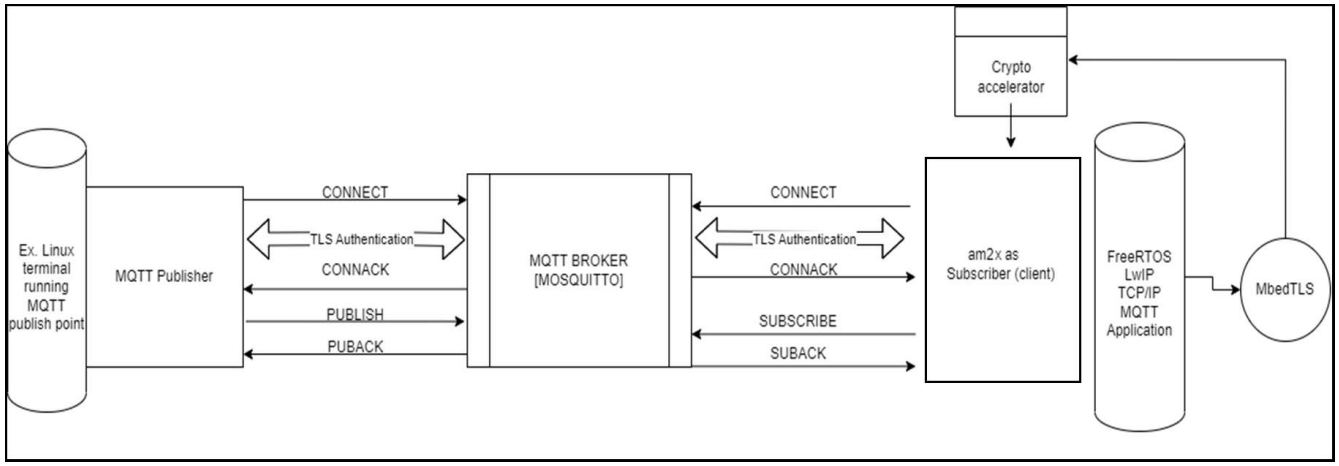

图 3-3. MQTT + TLS 客户端工作概述

重要声明和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做出任何明示或暗示的担保，包括但不限于对适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他功能安全、信息安全、监管或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的应用。严禁对这些资源进行其他复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。您应全额赔偿因在这些资源的使用中对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，TI 对此概不负责。

TI 提供的产品受 TI 的销售条款或 ti.com 上其他适用条款/TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

邮寄地址：Texas Instruments, Post Office Box 655303, Dallas, Texas 75265