应用笔记

使用非对称PKI的节点验证示例

ATECC508A

简介

node-auth-basic.atsln项目是一个一体化示例，它演示了使用CryptoAuthentication™器件（例如，ATECC508A）的公钥、非对称技术的节点验证序列的各个阶段。

将演示的各个节点验证阶段如下：

- 使用设备和签署者证书及密钥配置ATECC508A
- 通过存储在ATECC508A中的数据重建X.509证书
- 链验证——验证设备证书到可信根（Root of Trust，RoT）的链接
- 向设备发送质询
- 设备对质询进行签名
- 验证已签名质询的可靠性

概述

链验证阶段和设备质询/签名验证阶段的组合结果指示节点是否可靠并且可证明其是否为原始的OEM设备。它还举例说明了如何配置设备以保存X.509证书的关键数据。

先决条件

- 软件：
  - Atmel Studio 6.2或7.0
- 硬件：
  - SAM D21 Xplained Pro评估工具包
  - AT88CK101开发板（带插座）或
    CryptoAuth Xplained Pro评估和开发工具包

将CryptoAuthXplained Pro工具包插入SAM D21 Ext1或Ext2接头。SAM D21 Xplained Pro工具包的I^2C引脚自动连接CryptoAuth Xplained Pro。随后搭配使用该项目所需的固件，该示例即可运行。

目录

1 如果迫不及待，该从哪里着手？

对于迫不及待的读者来说，可以从node_auth.c开始阅读各个阶段的实现代码。该代码使用CryptoAuthLib库（一种便携式设备驱动程序）与ATECC508A通信。通过此项目示例，可从头到尾了解整个序列（包括最低的驱动程序级）。

本示例的HTML文档可以在node-auth-basic/docs目录下找到。使用浏览器加载index.html可查看node-auth-basic项目的文档。

CryptoAuthLib（CryptoAuthentication器件的内核加密库）的HTML文档可以在node-auth-basic/src/cryptoauthlib/docs/下找到。在浏览器中加载index.html即可查看CryptoAuthLib的API文档。

2 什么是节点？

本用例中的“节点”指的是要验证的设备。它可能是一个配件，甚至是网络中的一个传感器。

3 “一体化”是什么意思？

“一体化”意味着这些阶段往往不在同一个器件上执行。例如，节点可能是无线网络上的6LoWPAN设备，而主机位于远程数据中心。但是，也存在所有运行时阶段在同一主机上执行的用例。例如，在诸如打印机/打印机墨盒之类的消耗品用例中，打印机中的主机将执行本文所述的所有阶段，ATECC508A位于与主机直接电接触的墨盒中。

使用“一体化”示例，将非常便于观察所有角色如何在类似于打印机/打印机墨盒用例的系统中一起工作。一体化最大限度地减少了硬件，通过单一工具Atmel Studio跟踪所有代码路径从未如此简单。

本示例将明确区分在每个阶段所充当的角色。

4 本示例中演示了哪些角色？

一体化示例演示了以下角色：

- **配置者** 该角色负责配置和编程ATECC508A以供运行时使用。
- **客户端** 待验证的设备，如配件。
- **主机** 将执行验证步骤以保证设备可靠性的器件。

5 验证阶段

5.1 配置

通常情况下，ATECC508A的出厂配置阶段在工厂中进行，本文对这一阶段进行介绍，旨在演示在设备中存储证书的基本过程。

5.2 重建

重建是一种通过ATECC508A中存储的少量数据（这些数据作为证书的一部分动态创建）重新组合成完全有效的X.509有效证书的方法。

5.3 链验证，可信根

椭圆曲线数字签名算法（Elliptic Curve Digital Signature Algorithm，ECDSA）验证可信根（Root of Trust，RoT）是完整验证过程的其中一个阶段，用于确保该器件已正确签署到制造商的证书链中。如果某个证书无效或包含不正确签名者的签名或公钥，此链将失效。

该验证过程可防止攻击者在RoT链上伪造一个证书。

5.4 质询签名验证

ECDSA质询签名验证通常是在主机向待加密签名的ATECC508A发送的质询（随机数）被签名后进行。签名中包含由ATECC508A安全保存的私钥，该私钥无法通过硬件读取。随后会使用设备的公钥、签名和质询数据本身验证质询的签名。所有验证完成后，即可确定设备是否可靠，主机可以根据该结果执行适当的过程。

对主机发送的随机质询签名可证明设备确实拥有与其证书的公钥相关联的私钥。

5.5 编译示例源代码

如果使用Atmel Studio 6.2，请加载项目文件： `node-auth-basic_6_2.atsln`

如果使用Atmel Studio 7.0，请加载项目文件： `node-auth-basic.atsln`

项目加载完成后，立即使用***Build***（编译）菜单下的***Rebuild Solution***（重新编译解决方案）进行编译。使用标准Atmel Studio器件编程工具刷新SAM D21 Xplained Pro工具包。

5.6 使用节点验证基本示例

SAM D21 Xplained Pro上有两个USB端口。其中一个标有“EDBG USB”，用于通过Atmel Studio将代码写入MCU的闪存中。第二个USB端口标有“目标USB”，这是一个CDC USB端口，用于示例的控制台接口。

1. 将主机计算机连接到EDBG USB进行编程。
2. 将主机计算机连接到目标USB CDC端口，查看可用于在示例完成编程后对其进行演示的控制台接口。
3. 在主机上使用终端程序，并将其连接到SAM D21 Xplained Pro的虚拟COMM端口，当目标USB CDC端口插入PC、Linux®或OS X机器时应创建该虚拟端口。这一步会因计算机和操作系统的不同而有所差异。

通信参数如下：

- 115,200波特
- 8位字
- 无奇偶校验
- 1个停止位

帮助命令

连接到串行USB后，键入`help`，命令行控制台将如下所示：

```
$ help
Usage:
client-provision  - Configure and load certificate data onto ATECC device.
client-build      - Read certificate data off ATECC device and rebuild full
signer and device certificates.
host-chain-verify - Verify the certificate chain from the client.
host-gen-chal     - Generate challenge for the client.
client-gen-resp   - Generate response to challenge from host.
host-verify-resp  - Verify the client response to the challenge.
Utility functions:
lockstat - zone lock status
lockcfg  - lock config zone
lockdata - lock data and OTP zones
info     - get the chip revision
sernum   - get the chip serial number

$
```

检查ATECC508A连接

使用命令控制台时，可通过info或sernum命令来显示器件的版本和序列号。这些都是很好的测试，可确保您的主板与ATECC508A之间能够相互通信。

以下是预期的示例会话。当然，您的序列号会有所不同。

```
$ info
revision:
00 00 50 00
$ sernum
serial number:
01 23 61 12 D9 2C A5 71 EE
$
```

必须成功完成此步骤才能继续下一步。

如果显示的版本或序列号不一致，请检查与CryptoAuthXplained Pro扩展板之间的连接，或者检查与连接到SAM D21 Xplained Pro工具包I^2C引脚的带插座顶板之间的连接。

步骤1　　配置ATECC508A：client-provision

输入以下命令：client-provision

该步骤可一次性在ATECC508A中生成密钥并构建稍后完成验证步骤所需的证书。在此步骤中创建和存储的证书包括设备的证书和签名者的证书。

该命令完成后，所有证书和密钥将立即存储并锁定在设备中。此后不能更改该设备。

示例：　client-provision会话

采用client-provision的示例会话可能会如下所示。不必纠结其中显示的确切字节，实际情况会有所不同；关键在于可以看到各种组件已经创建并含有数据。

```
Signer CA Public Key:
02 54 9E 50 2F 7C 13 1E C5 DA 7A 8B BF 5E 0D 05
E1 3D 8E 11 F4 F1 04 D2 F6 CE 41 44 FA 40 E6 D4
02 3C A0 80 30 B1 DE F1 4A A7 CE A3 FF 12 4B 4B
A5 91 E0 F1 59 EF 67 A9 68 E5 CC 5C 0B FD E8 7A
Signer Public Key:
A3 AC C0 2F 35 17 15 08 68 B1 10 43 24 F9 EA 30
17 2C B1 11 AB A1 F0 B5 0B 4B 85 77 2B F3 14 08
70 C0 69 8E AF AA 6A 58 F9 8E 22 0F 3A 9E F8 35
C0 6A 5D FB C5 25 F4 56 5A A7 AB A9 E9 B1 44 E6
Device Public Key:
B9 17 F9 9F BA A0 AF 3C 67 61 B8 DB D8 2F 8E 6B
C1 CB D0 CF 87 82 08 0E 2B D3 EC EF E8 E9 C5 3B
E2 1C 2E 5D CC A1 92 A5 A1 22 68 EA FF 94 68 F5
C0 54 DD 32 40 F9 F6 C2 9B AF 0D 46 36 EC 5F 26
Signer Certificate:
30 82 01 B1 30 82 01 57 A0 03 02 01 02 02 03 40
```

```
C4 8B 30 0A 06 08 2A 86 48 CE 3D 04 03 02 30 36
31 10 30 0E 06 03 55 04 0A 0C 07 45 78 61 6D 70
6C 65 31 22 30 20 06 03 55 04 03 0C 19 45 78 61
6D 70 6C 65 20 41 54 45 43 43 35 30 38 41 20 52
6F 6F 74 20 43 41 30 1E 17 0D 31 34 30 38 30 32
32 30 30 30 30 30 5A 17 0D 33 34 30 38 30 32 32
30 30 30 30 30 5A 30 3A 31 10 30 0E 06 03 55 04
0A 0C 07 45 78 61 6D 70 6C 65 31 26 30 24 06 03
55 04 03 0C 1D 45 78 61 6D 70 6C 65 20 41 54 45
43 43 35 30 38 41 20 53 69 67 6E 65 72 20 43 34
38 42 30 59 30 13 06 07 2A 86 48 CE 3D 02 01 06
08 2A 86 48 CE 3D 03 01 07 03 42 00 04 A3 AC C0
2F 35 17 15 08 68 B1 10 43 24 F9 EA 30 17 2C B1
11 AB A1 F0 B5 0B 4B 85 77 2B F3 14 08 70 C0 69
8E AF AA 6A 58 F9 8E 22 0F 3A 9E F8 35 C0 6A 5D
FB C5 25 F4 56 5A A7 AB A9 E9 B1 44 E6 A3 50 30
4E 30 0C 06 03 55 1D 13 04 05 30 03 01 01 FF 30
1D 06 03 55 1D 0E 04 16 04 14 BB 5C 3D F7 4D 4C
93 D4 2B 50 D1 7F B3 23 C3 3A B0 2C 27 BA 30 1F
06 03 55 1D 23 04 18 30 16 80 14 14 B0 97 8A 1D
57 50 FF 52 F9 DF A8 90 60 77 60 C5 3C 6B 50 30
0A 06 08 2A 86 48 CE 3D 04 03 02 03 48 00 30 45
02 21 00 FB 08 10 99 B3 F0 A8 E5 D5 19 3F 1A A2
20 94 06 A1 63 D9 4A CE 18 6A 80 C6 6A E7 91 42
6C 58 7D 02 20 46 85 5F 9D 71 F2 B9 48 84 75 2E
49 2F D7 58 AD 1B EB BD 36 A5 74 64 2B 6B EA 02
26 5A 72 13 3F
Device Certificate:
30 82 01 8A 30 82 01 30 A0 03 02 01 02 02 0A 40
01 23 6F 12 D9 2C A5 71 EE 30 0A 06 08 2A 86 48
CE 3D 04 03 02 30 3A 31 10 30 0E 06 03 55 04 0A
0C 07 45 78 61 6D 70 6C 65 31 26 30 24 06 03 55
04 03 0C 1D 45 78 61 6D 70 6C 65 20 41 54 45 43
43 35 30 38 41 20 53 69 67 6E 65 72 20 43 34 38
42 30 1E 17 0D 31 35 30 39 30 33 32 31 30 30 30
30 5A 17 0D 33 35 30 39 30 33 32 31 30 30 30 30
5A 30 35 31 10 30 0E 06 03 55 04 0A 0C 07 45 78
61 6D 70 6C 65 31 21 30 1F 06 03 55 04 03 0C 18
45 78 61 6D 70 6C 65 20 41 54 45 43 43 35 30 38
41 20 44 65 76 69 63 65 30 59 30 13 06 07 2A 86
48 CE 3D 02 01 06 08 2A 86 48 CE 3D 03 01 07 03
42 00 04 B9 17 F9 9F BA A0 AF 3C 67 61 B8 DB D8
2F 8E 6B C1 CB D0 CF 87 82 08 0E 2B D3 EC EF E8
E9 C5 3B E2 1C 2E 5D CC A1 92 A5 A1 22 68 EA FF
94 68 F5 C0 54 DD 32 40 F9 F6 C2 9B AF 0D 46 36
EC 5F 26 A3 23 30 21 30 1F 06 03 55 1D 23 04 18
30 16 80 14 BB 5C 3D F7 4D 4C 93 D4 2B 50 D1 7F
B3 23 C3 3A B0 2C 27 BA 30 0A 06 08 2A 86 48 CE
3D 04 03 02 03 48 00 30 45 02 20 35 96 2E 3F F4
1A 3A DA E7 6F E1 FE 9D 7A 83 BE 36 FA 06 C5 01
79 55 F2 2C 8C FE 1D 43 38 19 CC 02 21 00 E8 53
87 83 A6 98 21 8E 43 A0 08 73 B3 FD B4 4B 7E 1C
EC FB 61 33 52 59 99 DF B1 E1 79 3E D7 8B
$
```

步骤2　　读取ATECC508A证书：client-build

输入以下命令：client-build

client-build可从ATECC508A读取证书数据并将其重建为X.509 DER格式的证书。对于本演示，您不必解析完整的证书；演示代码将使用X.509 DER格式来执行校验和验证步骤。

示例：　典型的client-build会话将如下所示。

您可以将此输出与客户端配置步骤中显示的证书进行比较。结果应当相同。在此步骤中，证书数据从设备读取，重建后应与配置设备时的预期数据一致。

```
  CLIENT: Rebuilt Signer Certificate:
  30 82 01 B1 30 82 01 57 A0 03 02 01 02 02 03 40
  C4 8B 30 0A 06 08 2A 86 48 CE 3D 04 03 02 30 36
  31 10 30 0E 06 03 55 04 0A 0C 07 45 78 61 6D 70
  6C 65 31 22 30 20 06 03 55 04 03 0C 19 45 78 61
  6D 70 6C 65 20 41 54 45 43 43 35 30 38 41 20 52
  6F 6F 74 20 43 41 30 1E 17 0D 31 34 30 38 30 32
  32 30 30 30 30 30 5A 17 0D 33 34 30 38 30 32 32
  30 30 30 30 30 5A 30 3A 31 10 30 0E 06 03 55 04
0A 0C 07 45 78 61 6D 70 6C 65 31 26 30 24 06 03
55 04 03 0C 1D 45 78 61 6D 70 6C 65 20 41 54 45
43 43 35 30 38 41 20 53 69 67 6E 65 72 20 43 34
38 42 30 59 30 13 06 07 2A 86 48 CE 3D 02 01 06
08 2A 86 48 CE 3D 03 01 07 03 42 00 04 A3 AC C0
2F 35 17 15 08 68 B1 10 43 24 F9 EA 30 17 2C B1
11 AB A1 F0 B5 0B 4B 85 77 2B F3 14 08 70 C0 69
8E AF AA 6A 58 F9 8E 22 0F 3A 9E F8 35 C0 6A 5D
FB C5 25 F4 56 5A A7 AB A9 E9 B1 44 E6 A3 50 30
4E 30 0C 06 03 55 1D 13 04 05 30 03 01 01 FF 30
1D 06 03 55 1D 0E 04 16 04 14 BB 5C 3D F7 4D 4C
93 D4 2B 50 D1 7F B3 23 C3 3A B0 2C 27 BA 30 1F
06 03 55 1D 23 04 18 30 16 80 14 14 B0 97 8A 1D
57 50 FF 52 F9 DF A8 90 60 77 60 C5 3C 6B 50 30
0A 06 08 2A 86 48 CE 3D 04 03 02 03 48 00 30 45
02 21 00 FB 08 10 99 B3 F0 A8 E5 D5 19 3F 1A A2
20 94 06 A1 63 D9 4A CE 18 6A 80 C6 6A E7 91 42
6C 58 7D 02 20 46 85 5F 9D 71 F2 B9 48 84 75 2E
49 2F D7 58 AD 1B EB BD 36 A5 74 64 2B 6B EA 02
26 5A 72 13 3F
 CLIENT: Rebuilt Device Certificate:
30 82 01 8A 30 82 01 30 A0 03 02 01 02 02 0A 40
01 23 6F 12 D9 2C A5 71 EE 30 0A 06 08 2A 86 48
CE 3D 04 03 02 30 3A 31 10 30 0E 06 03 55 04 0A
0C 07 45 78 61 6D 70 6C 65 31 26 30 24 06 03 55
04 03 0C 1D 45 78 61 6D 70 6C 65 20 41 54 45 43
43 35 30 38 41 20 53 69 67 6E 65 72 20 43 34 38
42 30 1E 17 0D 31 35 30 39 30 33 32 31 30 30 30
30 5A 17 0D 33 35 30 39 30 33 32 31 30 30 30 30
5A 30 35 31 10 30 0E 06 03 55 04 0A 0C 07 45 78
61 6D 70 6C 65 31 21 30 1F 06 03 55 04 03 0C 18
45 78 61 6D 70 6C 65 20 41 54 45 43 43 35 30 38
```

```
41 20 44 65 76 69 63 65 30 59 30 13 06 07 2A 86
48 CE 3D 02 01 06 08 2A 86 48 CE 3D 03 01 07 03
42 00 04 B9 17 F9 9F BA A0 AF 3C 67 61 B8 DB D8
2F 8E 6B C1 CB D0 CF 87 82 08 0E 2B D3 EC EF E8
E9 C5 3B E2 1C 2E 5D CC A1 92 A5 A1 22 68 EA FF
94 68 F5 C0 54 DD 32 40 F9 F6 C2 9B AF 0D 46 36
EC 5F 26 A3 23 30 21 30 1F 06 03 55 1D 23 04 18
30 16 80 14 BB 5C 3D F7 4D 4C 93 D4 2B 50 D1 7F
B3 23 C3 3A B0 2C 27 BA 30 0A 06 08 2A 86 48 CE
3D 04 03 02 03 48 00 30 45 02 20 35 96 2E 3F F4
1A 3A DA E7 6F E1 FE 9D 7A 83 BE 36 FA 06 C5 01
79 55 F2 2C 8C FE 1D 43 38 19 CC 02 21 00 E8 53
87 83 A6 98 21 8E 43 A0 08 73 B3 FD B4 4B 7E 1C
EC FB 61 33 52 59 99 DF B1 E1 79 3E D7 8B
$
```

步骤3　　验证证书链：host-chain-verify

输入以下命令：host-chain-verify

host-chain-verify可从ATECC508A中获取设备证书和签名者证书、重建证书，然后执行链验证，链验证将验证设备证书是否有效以及是否已经签署到RoT链中。

示例：　　典型的host-chain-verify会话将如下所示：

```
$ host-chain-verify
HOST: Signer certificate verified against signer certificate authority (CA) public key!
HOST: Device certificate verified against signer public key!
```

步骤4　　从主机生成质询：host-gen-chal

输入以下命令：host-gen-chal

host-gen-chal可生成一个随机质询，并要求ATECC508A使用存储在ATECC508A中与设备证书相对应的私钥进行签名。

这是典型“质询/响应”模式的前半部分。在接收到响应之后（步骤5），可以执行ECDSA验证，该验证通过数学方式确定签名是否有效。

示例：　　质询将如下所示：

```
$ host-gen-chal
HOST: Generated challenge:
14 84 E8 89 41 D5 9A 1C AD 1F 68 44 3A 09 C6 45
30 BF 27 38 D2 28 56 B7 DD D6 98 CF 92 AB 3D 69
```

步骤5　　生成对质询的响应（签名）：client-gen-resp

输入以下命令：client-gen-resp

client-gen-resp可生成在步骤4中执行的质询的签名。该命令会要求ATECC508A对质询进行签名，并返回其生成的签名。此签名将用于接下来的验证步骤。

示例：　　签名的生成过程将如下所示：

```
$ client-gen-resp
CLIENT: Calculated response to host challenge:
BB BD 18 73 C3 88 86 E7 86 4A 53 CF 8F 18 4D EC
1A 39 A2 B9 FC 0B FE 73 CE 51 42 0C FB 81 26 F9
63 C1 A0 AF A8 67 58 FB 3B 9D 19 6B FE 86 98 47
0C 13 C9 95 8D 37 C9 47 57 61 A0 F7 D4 52 42 45
```

步骤6　　验证签名：host-verify-resp

输入以下命令：host-verify-resp

host-verify-resp可执行ECDSA验证以确定签名是否有效。ECDSA验证需要三部分数据：

- 器件的公钥。
- 发送给设备令其签名的质询。
- 质询的签名。

如果设备通过ECDSA验证步骤的验证，则可证明其具有与其设备证书中的公钥相关联的私钥，并签署到证书链中。具体来说就是，设备已证明其拥有公钥，并且如果其具有相同公钥的证书通过了链验证，则认为设备通过完全认证，是一个可靠的OEM设备。

示例：　　设备验证最后一步：

```
$ host-verify-resp
CLIENT: Calculated response to host challenge:
BB BD 18 73 C3 88 86 E7 86 4A 53 CF 8F 18 4D EC
1A 39 A2 B9 FC 0B FE 73 CE 51 42 0C FB 81 26 F9
63 C1 A0 AF A8 67 58 FB 3B 9D 19 6B FE 86 98 47
0C 13 C9 95 8D 37 C9 47 57 61 A0 F7 D4 52 42 45
HOST: Device public key from certificate:
B9 17 F9 9F BA A0 AF 3C 67 61 B8 DB D8 2F 8E 6B
C1 CB D0 CF 87 82 08 0E 2B D3 EC EF E8 E9 C5 3B
E2 1C 2E 5D CC A1 92 A5 A1 22 68 EA FF 94 68 F5
C0 54 DD 32 40 F9 F6 C2 9B AF 0D 46 36 EC 5F 26
HOST: Device response to challenge verified!
```

6 版本历史

文档版本	日期	备注
8983A	2015年9月	文档初始版本。

请注意以下有关Microchip 器件代码保护功能的要点：

- Microchip 的产品均达到 Microchip 数据手册中所述的技术指标。
- Microchip 确信：在正常使用的情况下， Microchip 系列产品是当今市场上同类产品中最安全的产品之一。
- 目前，仍存在着恶意、甚至是非法破坏代码保护功能的行为。就我们所知，所有这些行为都不是以 Microchip 数据手册中规定的操作规范来使用 Microchip 产品的。这样做的人极可能侵犯了知识产权。
- Microchip 愿与那些注重代码完整性的客户合作。
- Microchip 或任何其他半导体厂商均无法保证其代码的安全性。代码保护并不意味着我们保证产品是 “牢不可破”的。

代码保护功能处于持续发展中。Microchip 承诺将不断改进产品的代码保护功能。任何试图破坏 Microchip 代码保护功能的行为均可视为违反了《数字器件千年版权法案（Digital Millennium Copyright Act）》。如果这种行为导致他人在未经授权的情况下，能访问您的软件或其他受版权保护的成果，您有权依据该法案提起诉讼，从而制止这种行为。

Microchip 位于美国亚利桑那州 Chandler 和 Tempe 与位于俄勒冈州 Gresham的全球总部、设计和晶圆生产厂及位于美国加利福尼亚州和印度的设计中心均通过了 ISO/TS-16949:2009 认证。 Microchip 的 PIC® MCU 与 dsPIC® DSC、KEELOQ® 跳码器件、串行 EEPROM、单片机外设、非易失性存储器和模拟产品严格遵守公司的质量体系流程。此外，Microchip 在开发系统的设计和生产方面的质量体系也已通过了 ISO 9001:2000 认证。

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

10/25/17

全球销售及服务网点

美洲

公司总部 Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 1-480-792-7200
Fax: 1-480-792-7277
技 术 支 持 ：
http://www.microchip.com/support
网址：www.microchip.com

亚特兰大 Atlanta
Duluth, GA
Tel: 1-678-957-9614
Fax: 1-678-957-1455

奥斯汀 Austin, TX
Tel: 1-512-257-3370

波士顿 Boston
Westborough, MA
Tel: 1-774-760-0087
Fax: 1-774-760-0088

芝加哥 Chicago
Itasca, IL
Tel: 1-630-285-0071
Fax: 1-630-285-0075

达拉斯 Dallas
Addison, TX
Tel: 1-972-818-7423
Fax: 1-972-818-2924

底特律 Detroit
Novi, MI
Tel: 1-248-848-4000

休斯敦 Houston, TX
Tel: 1-281-894-5983

印第安纳波利斯 Indianapolis
Noblesville, IN
Tel: 1-317-773-8323
Fax: 1-317-773-5453
Tel: 1-317-536-2380

洛杉矶 Los Angeles
Mission Viejo, CA
Tel: 1-949-462-9523
Fax: 1-949-462-9608
Tel: 1-951-273-7800

罗利 Raleigh, NC
Tel: 1-919-844-7510

纽约 New York, NY
Tel: 1-631-435-6000

圣何塞 San Jose, CA
Tel: 1-408-735-9110
Tel: 1-408-436-4270

加拿大多伦多 Toronto
Tel: 1-905-695-1980
Fax: 1-905-695-2078

亚太地区

中国 - 北京
Tel: 86-10-8569-7000

中国 - 成都
Tel: 86-28-8665-5511

中国 - 重庆
Tel: 86-23-8980-9588

中国 - 东莞
Tel: 86-769-8702-9880

中国 - 广州
Tel: 86-20-8755-8029

中国 - 杭州
Tel: 86-571-8792-8115

中国 - 南京
Tel: 86-25-8473-2460

中国 - 青岛
Tel: 86-532-8502-7355

中国 - 上海
Tel: 86-21-3326-8000

中国 - 沈阳
Tel: 86-24-2334-2829

中国 - 深圳
Tel: 86-755-8864-2200

中国 - 苏州
Tel: 86-186-6233-1526

中国 - 武汉
Tel: 86-27-5980-5300

中国 - 西安
Tel: 86-29-8833-7252

中国 - 厦门
Tel: 86-592-238-8138

中国 - 香港特别行政区
Tel: 852-2943-5100

中国 - 珠海
Tel: 86-756-321-0040

台湾地区 - 高雄
Tel: 886-7-213-7830

台湾地区 - 台北
Tel: 886-2-2508-8600

台湾地区 - 新竹
Tel: 886-3-577-8366

亚太地区

澳大利亚 Australia - Sydney
Tel: 61-2-9868-6733

印度 India - Bangalore
Tel: 91-80-3090-4444

印度 India - New Delhi
Tel: 91-11-4160-8631

印度 India - Pune
Tel: 91-20-4121-0141

日本 Japan - Osaka
Tel: 81-6-6152-7160

日本 Japan - Tokyo
Tel: 81-3-6880-3770

韩国 Korea - Daegu
Tel: 82-53-744-4301

韩国 Korea - Seoul
Tel: 82-2-554-7200

马来西亚 Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906

马来西亚 Malaysia - Penang
Tel: 60-4-227-8870

菲律宾Philippines - Manila
Tel: 63-2-634-9065

新加坡Singapore
Tel: 65-6334-8870

泰国 Thailand - Bangkok
Tel: 66-2-694-1351

越南 Vietnam - Ho Chi Minh
Tel: 84-28-5448-2100

欧洲

奥地利Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

丹麦 Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

芬兰 Finland - Espoo
Tel: 358-9-4520-820

法国 France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

德国 Germany - Garching
Tel: 49-8931-9700

德国 Germany - Haan
Tel: 49-2129-3766400

德国 Germany - Heilbronn
Tel: 49-7131-67-3636

德国 Germany - Karlsruhe
Tel: 49-721-625370

德国 Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

德国 Germany - Rosenheim
Tel: 49-8031-354-560

以色列 Israel - Ra'anana
Tel: 972-9-744-7705

意大利Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

意大利Italy - Padova
Tel: 39-049-7625286

荷兰 Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

挪威 Norway - Trondheim
Tel: 47-7289-7561

波兰 Poland - Warsaw
Tel: 48-22-3325737

罗马尼亚 Romania - Bucharest
Tel: 40-21-407-87-50

西班牙Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

瑞典 Sweden - Gothenberg
Tel: 46-31-704-60-40

瑞典 Sweden - Stockholm
Tel: 46-8-5090-4654

英国 UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820