# CoAP 协议隐蔽通道分析及安全建议

姬国珍，康荣保，田学成，孙宁天，张　蕾
（中国电子科技集团公司第三十研究所，四川　成都　610041）

**摘　要：**物联网（IoT）设备的快速普及为关键基础设施网络带来了新的安全挑战，利用物联网通信协议中的隐蔽通道泄露敏感数据，实现远程控制对关键基础设施网络产生严重威胁。通过分析物联网 CoAP 协议隐蔽信道构建方法，并利用协议不同字段构建了多个隐蔽通道。在实验环境中验证了隐蔽通道的隐蔽性带来的安全挑战。该研究揭示了物联网环境下的安全威胁，为物联网的安全防护提供技术支撑。

**关键词：**物联网；隐蔽信道；CoAP 协议；安全防护
**中图分类号：**TP309　　**文献标志码：**A　　**DOI：**10. 19358/j. issn. 2097 - 1788. 2026. 02. 002

**中文引用格式：**姬国珍，康荣保，田学成，等. CoAP 协议隐蔽通道分析及安全建议［J］. 网络安全与数据治理，2026，45(2)：12 - 19.
**英文引用格式：**Ji Guozhen，Kang Rongbao，Tian Xuecheng，et al. Analysis and security recommendations for covert channels using the CoAP protocol［J］. Cyber Security and Data Governance，2026，45(2)：12 - 19.

## Analysis and security recommendations for covert channels using the CoAP protocol

Ji Guozhen，Kang Rongbao，Tian Xuecheng，Sun Ningtian，Zhang Lei
（The 30th Research Institute of China Electronics Technology Group Corporation，Chengdu 610041，China）

**Abstract：**The rapid proliferation of Internet of Things（IoT）devices has brought new security challenges to critical infrastructure networks. The use of covert channels in IoT communication protocols to leak sensitive data and enable remote control poses a serious threat to critical infrastructure networks. This paper analyzes the method of constructing covert channels in the IoT CoAP protocol and builds multiple covert channels using different fields of the protocol. The security challenges posed by the covertness of these channels are verified in an experimental environment. This research has revealed the security threats in the IoT environment，providing technical support for IoT security protection.

**Key words：**Internet of Things（IoT）；covert channel；CoAP protocol；security protection



## 0　引言

随着物联网（Internet of Things，IoT）技术的飞速发展，物联网已从智能家居延伸至能源电网、交通管控、工业制造、水利等关键基础设施网络，构建起“万物互联”的智能化运行体系。关键基础设施作为国家核心战略资源，承载着社会运转与经济发展的核心功能，而物联网设备的广泛接入打破了传统关键基础设施网络的封闭性，使其面临更为复杂多元的安全威胁。物联网终端的异构性、轻量化、低功耗特性，以及设备间通信协议的多样性，为恶意攻击提供了可乘之机，尤其在协议安全防护存在短板的场景下，安全风险被进一步放大。其中，隐蔽通道作为一种规避常规检测的非授权通信手段，凭借对合法协议、系统资源的滥用能力，逐渐成为攻击者实施数据窃取、指令渗透、长期驻留的核心工具。隐蔽通道的发展趋势逐渐从互联网延伸到物联网领域，对关键基础设施网络的保密性、完整性与可用性构成严重威胁。

本文通过研究物联网应用约束协议[1]（Constrained Application Protocol，CoAP）隐蔽通道构建技术，分析 CoAP 协议的安全性，为关键基础设施的安全防护提供技术支撑。

## 1　隐蔽通道概念及发展趋势

### 1.1　隐蔽通道概念分类

网络隐蔽通道[2-3]（Network Covert Channel，NCC）是指在正常的网络通信过程中，利用协议或系统机制中未被充分利用的部分来秘密传输信息的一种技术。隐蔽通道能够穿越网络防火墙和入侵检测系统，是网

络安全领域的重要研究方向。

根据信息隐藏的方式和位置，网络隐蔽通道主要可以分为存储型隐蔽通道（Storage Covert Channel，SCC）和时间型隐蔽通道（Timing Covert Channel，TCC）两类[3]。存储型隐蔽通道是指修改网络数据包中的未使用字段或者自定义扩展字段、填充字段来存储有效载荷。时间型隐蔽通道是通过调制网络数据包的发送时间间隔来编码秘密数据。

### 1.2 隐蔽通道发展趋势

目前，国内外相关研究学者主要聚焦于以互联网协议为载体的隐蔽通道构建，针对工业控制网络、物联网的隐蔽通道研究相对较少。智能家居和智能工厂的广泛应用，为隐蔽信道的构建提供了充足的环境。在相关研究方面，郭蕊等人[1]提出利用 CoAP 协议参数序列构建隐蔽通道，通过组合资源编码方式进行隐蔽信息的传输。该隐蔽信道依赖服务器资源类型的多少，每个数据包传输比特较少。邓雨欣等人[2]通过对物联网消息队列遥测传输（Message Queuing Telemetry Transmission，MQTT）协议通信命令的不同分组编码信息实现了隐蔽通道构建。李凤华等人[3]对物联网、云计算以及卫星互联网等泛在网络环境下隐蔽通道的度量、构建和检测问题进行全面分析。Velinov 等人[4-5]描述了消息队列遥测传输 MQTT 协议的 7 个直接隐蔽通道和 6 个间接隐蔽通道的构建方法，并对它们进行评估和分类。常慧妍等人[6]对多媒体流应用程序型和实时在线游戏两类应用程序构建隐蔽信道进行了全面研究分析。与郭蕊等人提出的通过操控 CoAP 协议 Accept 选项的可选媒体类型实现隐蔽信息传输方式不同，本文通过利用 CoAP 协议的 Token、Options 中的 Uri-Path、Uri_Query、Block2 等选项字段以及 Payload 可选字段构建了加密认证的隐蔽通道实现信息传输。从隐蔽通信领域分析了 CoAP 协议的安全性，并给出了相应的安全建议。

### 1.3 网络隐蔽通道构建过程

构建网络隐蔽通道，首先需分析能够实现隐蔽通信的协议字段或者通信模式，然后对原始数据进行编码，将原始数据转换为协议运行的形式，同时为了增加隐蔽性，通常会对数据进行加密或混淆处理，防止简单的模式匹配发现异常。

对于时间型隐蔽通道来说，通过调整数据包发送的时间间隔来编码信息。对于存储型隐蔽通道来说，发送端可以在协议消息的有效载荷中嵌入秘密信息，将数据直接编码到请求或响应的有效载荷中。服务端从协议请求的有效载荷中提取并解码信息。

基于物联网通信协议的网络隐蔽通道构建过程如图 1 所示。

### 1.4 网络隐蔽通道的度量方法

在网络隐蔽通道的设计与评估中，隐蔽性、鲁棒性和传输效率是三个关键指标。它们各自关注不同的方面，共同决定了隐蔽通道的整体性能和适用场景。

（1）隐蔽性[7-9]：指的是隐蔽通道在正常网络流量中的不可检测性或难以被发现的程度。一个高隐蔽性的通道应尽量避免引起安全设备（如防火墙、入侵检测系统 IDS/IPS）或网络管理员的注意。

（2）鲁棒性[7]：是指隐蔽通道在面对各种干扰和变化时保持功能的能力。这包括应对网络拓扑变化、节点故障、带宽波动以及主动防御措施（如过滤规则更新）等的挑战。

（3）传输效率[7]：衡量的是隐蔽通道在单位时间内能够传输的有效数据量，通常以比特率（b/s）表示。它反映了通道的实际数据传输能力和资源利用率。

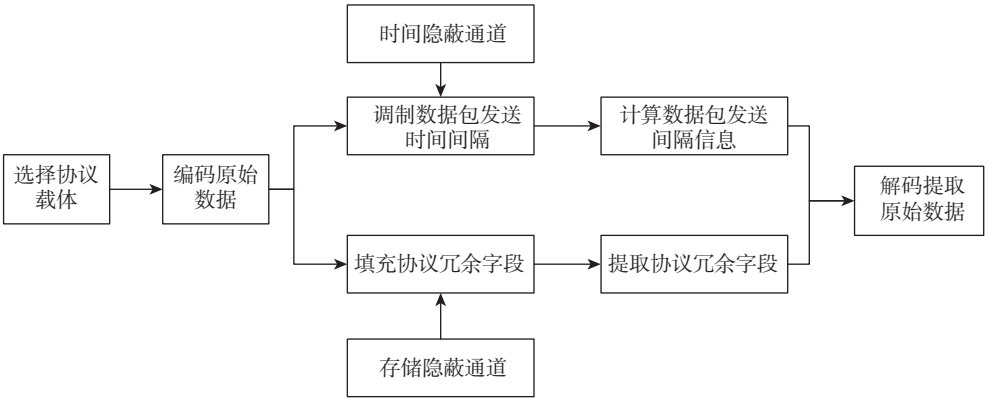



图 1　网络隐蔽通道构建过程

## 2 CoAP 协议分析

### 2.1 CoAP 协议介绍

CoAP 协议是一种专为物联网中的受限节点和网络设计的特殊 Web 传输协议。它不仅能够轻松转换为 HTTP 以便与 Web 无缝集成，同时还满足特定的要求，如多播支持、极低的开销和适用于受限环境的简洁特性。基于 REST 架构，该协议将网络中的各种对象视为资源。这些资源被唯一地分配了一个 URI（统一资源标识符）。资源之间的数据传输以 CoAP 消息包的形式交互。客户端请求某些资源，服务器根据消息类型确定是否确认信息作为响应。

CoAP 基于请求 - 响应模型，其主要方法类似于 HTTP（如 GET、POST、PUT 和 DELETE），并具有其独特的 Observe 方法。

### 2.2 CoAP 协议消息结构

CoAP 消息采用二进制格式或“0/1”格式编码。CoAP 消息包含头部和 Payload 负载两部分，头部和负载之间使用单字节 0xFF 隔开。CoAP 头部包含 4 B 的固定大小部分和 Token、Options 两个可选部分，Options 部分大小可变。负载也是可选部分。

CoAP 消息包其格式如图 2 所示，包含以下字段：

（1）Version：版本，字段大小为 2 bit，表示 CoAP 协议的版本。

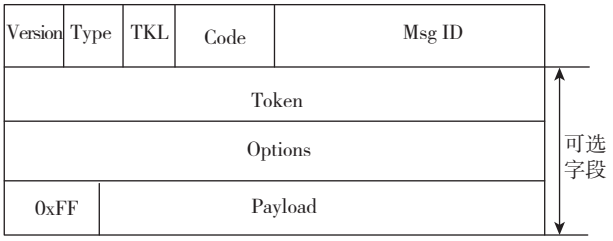



图 2　CoAP 协议消息结构

（2）Type：类型，字段大小为 2 bit。消息分为四种类型：可确认消息 CON、不可确认消息 NON、确认消息 ACK 和重置消息 RST，分别用位模式 00、01、10 和 11 表示。

（3）TKL（Token Len）：选项计数，字段大小为 4 bit，表示变长的 Token 字段（0 ~ 8 B）的长度。

（4）Code：代码，字段大小为 8 bit。8 bit 拆分为 3 bit 分类信息和 5 bit 详细信息。

（5）Msg ID：消息 ID，字段大小为 16 bit。它用于检测消息重复和消息类型。

（6）Token［可选］：令牌，字段的大小可变，范围为 0 ~ 8 B。它用于将响应与请求进行匹配。

（7）Options［可选］：CoAP 消息中的选项字段大小可变。它定义有效负载消息的类型。Options 字段的具体定义参见表 1。

表 1　CoAP 协议 Options 字段详细定义

| 选项编号 | 选项名称 | 数据类型 | 长度限制/B | 用途 |
|---|---|---|---|---|
| 1 | If-Match | opaque | 0 ~ 8 | 匹配资源的 ETag，用于条件请求 |
| 3 | Uri-Host | string | 1 ~ 255 | 指定目标服务器的主机名 / IP |
| 4 | ETag | opaque | 1 ~ 8 | 资源的实体标签，用于缓存验证、并发控制 |
| 5 | If-None-Match | empty | 0 | 条件请求：仅当资源不存在时执行 |
| 7 | Uri-Port | uint | 0 ~ 2 | 指定目标服务器的端口 |
| 8 | Location-Path | string | 0 ~ 255 | 用于响应中指定创建资源的路径 |
| 11 | Uri-Path | string | 0 ~ 255 | URI 的路径分段 |
| 12 | Content-Format | uint | 0 ~ 2 | 有效载荷的数据格式（如 0 = text/plain，40 = application/json） |
| 14 | Max-Age | uint | 0 ~ 4 | 响应的最大缓存时间 |
| 15 | Uri_Query | string | 0 ~ 255 | URI 的查询参数分段 |
| 17 | Accept | uint | 0 ~ 2 | 请求期望的响应数据格式 |
| 20 | Location-Query | string | 0 ~ 255 | 响应中指定创建资源的查询参数 |
| 23 | Block2 | uint | 0 ~ 3 | 用于块传输（Block-Wise），客户端请求分片数据时使用 |
| 27 | Block1 | uint | 0 ~ 3 | 用于块传输，服务器响应分片数据时使用 |
| 28 | Size2 | uint | 0 ~ 4 | 指示请求载荷的大小 |
| 35 | Proxy-Uri | string | 1 ~ 1 034 | 指定代理转发的完整 CoAP URI |
| 39 | Proxy-Scheme | string | 1 ~ 255 | 指定代理使用的协议方案 |
| 60 | Size1 | uint | 0 ~ 4 | 指示响应载荷的大小 |
| 128 ~ 255 | （预留） | — | — | 用于私有 / 实验用途（自定义扩展选项） |

（8）0xFF：CoAP 协议头部和负载部分之间的分隔符。

（8）Payload［可选］：与选项字段类似，有效负载字段的大小也可变。

### 2.3 CoAP 协议构建隐蔽通道可行性分析

CoAP 协议在电网实时监控、国防通信、航空等领域得到广泛的应用，用于传感器和执行器之间的通信。

（1）CoAP 协议可选 Token 字段，为 0 ~ 8 B。它用于将响应与请求进行匹配。单个数据包 Token 字段可以隐藏最多 8 个字节的信息。

（2）Options 字段定义了 18 个可选选项。其中多个选项可以作为隐蔽信息传输的载体。Payload 字段作为可选字段，可以传输大量数据。

（3）CoAP 协议作为轻量级物联网协议，在低功耗受限设备得到广泛应用，为信息隐藏提供了充足的信道。

### 2.4 CoAP 协议存储隐蔽通道构建方法

#### 2.4.1 Token 字段隐蔽通道构建方法

CoAP 协议头部长度为 4 个字节，头部之后是 Token 值，可以有 0 ~ 8 个字节，由 Token 长度字段指定。这个 Token 值用于将某个请求和对应的响应关联。一个数据包的 Token 值可以隐藏 8 个字节的数据，需要将待发送的隐蔽信息 msgData 进行编码后按照每 8 个字节为一组，需要发送的数据包数量为：

$$N = M/8 + 1 \tag{1}$$

其中，$N$ 为发送的数据包数量，$M$ 为编码后信息长度，“/”表示取整操作。

最后通过构造带 Token 参数的 GET 请求发送到服务器端指定地址。服务器端接收所有数据包，并且拼接 Token 字段，对拼接后的 Token 字段进行解码即可得到原始数据信息。

#### 2.4.2 Option 字段隐蔽通道构建

CoAP 定义了许多 Option，每个 Option 是一个 Type-Length-Value（TLV）结构，包含 Option Delta、Option Length 和 Option Value 三部分。CoAP 协议采用增量编码（Delta Encoding）以节省空间。Option Delta 表示当前 Option Number 与前一个 Option Number 的差值（第一个 Option Delta 初始化为 0）。Option Length 为该 Option 的 Value 部分的字节长度，如果 Length 字段值为 13 ~ 14，则后面跟 1 ~ 2 B 的扩展长度；若为 15，表示无效（保留）。Option Value 表示 Option 的具体内容。Option 字段的详细定义如表 1 所示。

其中，Options 字段中 Uri_Path、Uri_Query、Block、Proxy-Uri、Proxy-Scheme 选项均可以用来实现隐蔽通信，下面以前三个选项为例描述数据隐藏编码原理。

#### 2.4.3 利用 Uri_Path 编码隐藏数据

首先，用 Base64 安全编码对原始数据进行编码，将编码后的数据分为多段，每一段数据添加类似“api_”的前缀，并将修改后分段数据用路径连接符“/”进行拼接，最后将拼接后的参数和访问服务的地址进行组装，格式如下，将其作为 URI 参数。构造 GET 请求数据包发送给服务器端。

coap：//localhost：5683/api_JBSWY3DP/v1_L5BW6QKQ/device_L5BW65TF/sensor_OJ2CC

服务器端接收到 URI 参数后从分段路径中提取编码数据进行拼接，然后利用 Base64 编码方法解码即可恢复原始数据。

#### 2.4.4 利用 Uri_Query 编码隐藏数据

Uri_Query 编码隐藏数据的方法是将原始数据利用 Base64 安全编码后再按照事先约定好的长度分为多个数据段，如 ED1、ED2 等，作为查询参数 $p1$、$p2$ 等的参数，最后按照如下数据格式进行拼接。

coap：//localhost：5683/q？p1 = ED1&p2 = ED2

服务器端对接收到的 URI 参数中的查询参数进行拼接后，利用 Base64 安全解码即可恢复原始数据。

#### 2.4.5 利用 Block-wise 块传输编码隐藏数据

从上面的分析可以看出，Token 隐蔽通道传输隐蔽载荷信息较少，可以传输加密后的秘钥等数据，Uri_Path 和 Uri_Query 隐蔽数据通过 Base64 进行安全编码分段拼接到路径参数中，隐蔽性较强，适用于传输信息不大的场景，对于可执行文件、固件等大的信息适合采用 Block-wise 块传输方案。

Block 块传输隐藏数据的原理，是利用 CoAP 协议的 Block-wise 块传输机制来处理大数据量（超过 1 280 B），从而在其中实现数据隐蔽。当单个 CoAP 报文无法承载全部数据时（例如，语音数据超过网络最大传输单元 MTU），该机制会将数据分块传输。发送方将数据分割成多个块，每个块作为一个独立的 CoAP 消息发送，接收方则负责重组这些块以还原原始数据。Block2 用于实现服务器端到客户端的数据传输。Block-wise 块传输方案一般用于固件更新、语音数据上传。

Block-wise 块传输隐蔽通道构建方式是通过在客户端对隐藏数据进行编码转化为 Byte 后分为多个块，每个块分配一个编号，编号采用原始编号 + 偏移的方法。客户端构造 GET 请求，请求的 URI 地址类似以下格式：

{server_uri}? block = {params ['block_number']}

其中，server_uri 是服务器地址以及端口信息。将块编号隐藏在 URI 参数中以避免协议校验，对应的编码数据块作为 CoAP 协议的 Payload 发送给服务器端。服务器端从 URI 参数中提取 Block 块编号，解码偏移量，按照块编号排序拼接所有编码数据，最后完成对所有数据的解码。数据块大小用 $S$ 表示，其计算公式如下：

$$S = 2^{(N+4)} \tag{2}$$

其中，$N$ 为指数部分，取值范围为 0 ~ 6，不同 $N$ 值下对应数据块大小为 16 ~ 1 024 B。

#### 2.4.6 块传输编码数据加密认证

由于载荷数据在 Payload 字段很容易被包检测，因此改进的思路是对隐蔽数据进行加密传输，并增加校验功能。本文采用 AES-GCM 认证加密算法。利用 CoAP 协议的 CON/ACK 机制确保消息的可靠性。

AES-GCM（Advanced Encryption Standard-Galois/Counter Mode）是一种由 NIST（美国国家标准与技术研究院）在 2007 年提出的加密认证算法[10-11]，可同时提供机密性（Confidentiality）、完整性（Integrity）和真实性（Authenticity）。其另一个优点是输入的数据长度几乎没有限制，输出认证标签长度默认为 16 B。

AES-GCM 算法是由 AES-CTR（计数器模式加密）和 GMAC（Galois Message Authentication Code）组成，其中，AES-CTR 是一种流加密模式，将分组密码转换为流密码，不直接加密明文，而是对一个递增的计数器进行 AES 加密，并将加密结果与明文进行异或得到密文，提供机密性。GMAC 是 GCM 模式中用于消息认证的部分，是一种仅用于认证的消息验证码，生成认证标签，提供完整性以及认证功能。

加密和认证函数如式（3）、式（4）所示：

$$\text{Ciphertext} = \text{AES} - \text{CTR}(\text{Key}, \text{Nonce}, \text{Plaintext}) \tag{3}$$

$$\text{Auth Tag} = \text{GMAC}(\text{Key}, \text{Nonce}, \text{Ciphertext}, \text{AAD}) \tag{4}$$

其中：Key 为加密密钥，长度为 128/192/256 bit（16/24/32 B）；Nonce 为一次性随机数，推荐长度 12 B（96 bit），可变长；Plaintext 为待加密数据，任意长度；AAD（Associated Authenticated Data）为需认证但不加密的附加数据（如 CoAP 头部），可选，任意长度。

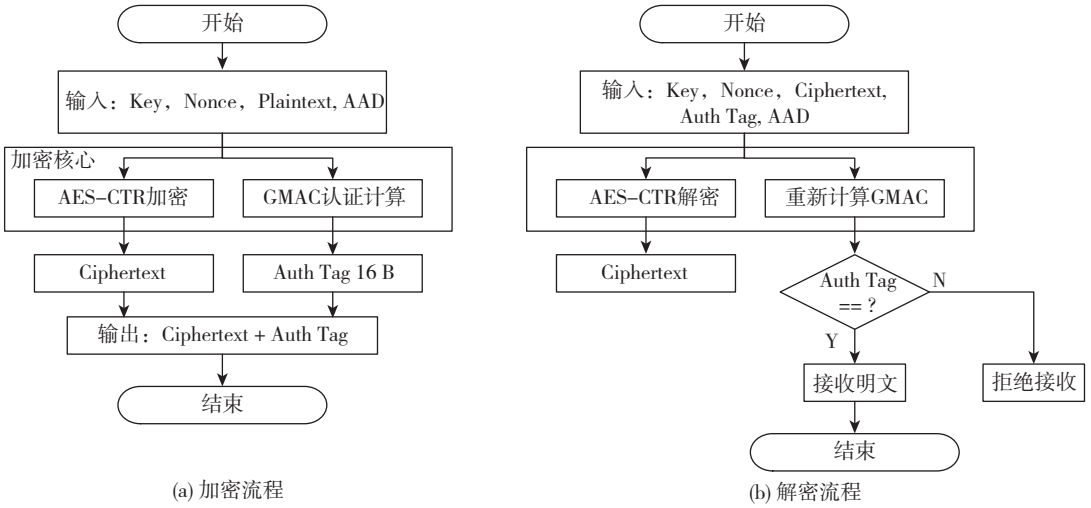
加密和解密流程如图 3 所示。

通过 AES-GCM 加密算法对编码数据进行加密，不仅确保了传输数据的机密性，还同时提供了完整性保护。该算法构建了“加密 + 校验”双重安全保障体系，同时在通信信息确认方面，构建了 CON/ACK 发送确认消息对确保消息可靠性。考虑到物联网环境内存小、带宽低、存储能力有限，因此减小 AES-GCM 算法中 Nonce 和 Tag 的长度开销，本文中 Nonce 长度设置为 12 B，Tag 长度设置为 16 B。通过 AES-GCM 加密认证算法保证协议传输内容的机密性和完整性。

## 3 实验结果与分析

### 3.1 实验环境设置

隐蔽通信场景如图 4 所示，实验环境主要由两台主机和一台交换机组成。其中，PC1 和 PC2 分别作为隐蔽通信的发送端和接收端。PC1 主机上布置 Windows 防火墙、360 安全卫士软件以及微软 Defender。PC2 主机上安装 Wireshark 抓包工具以及 Snort 入侵检测系统




(a) 加密流程　　(b) 解密流程

图 3　加密流程和解密流程

以检测隐蔽信道的隐蔽性。PC1 主机上运行隐蔽通道的发送端程序，PC2 主机运行隐蔽通道的接收端程序。

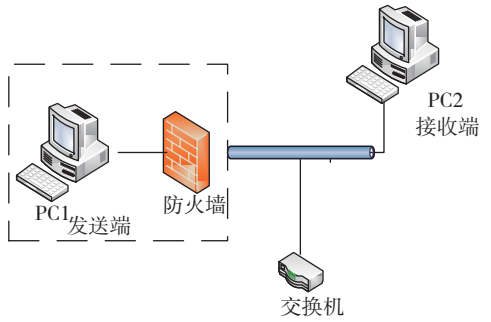



图 4　实验环境

### 3.2　实验结果及分析

图 5 ~ 图 7 分别是利用 CoAP 协议的 Token 字段，Uri_Path 和 Uri_Query，Block-wise 传输隐蔽信息的网络抓包数据。PC1 端主机上运行的发送程序能够绕过 Windows 防火墙，实现对 360 安全卫士和 Windows Defender 的免杀，同时，接收端的入侵检测系统 Snort 也没有报警，验证了该隐蔽通道的隐蔽性。

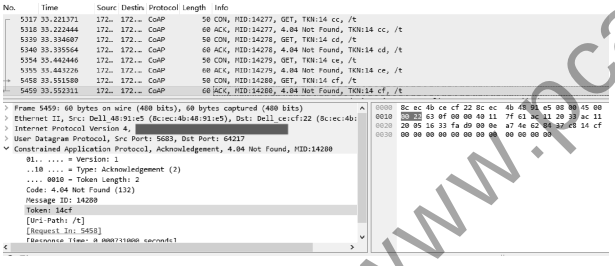


图 5　CoAP 协议 Token 字段传输隐蔽信息

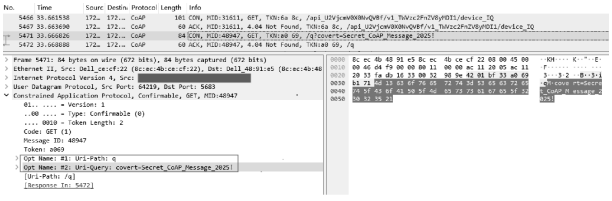


图 6　CoAP 协议 Uri_Path 以及 Uri_Query 传输隐蔽信息

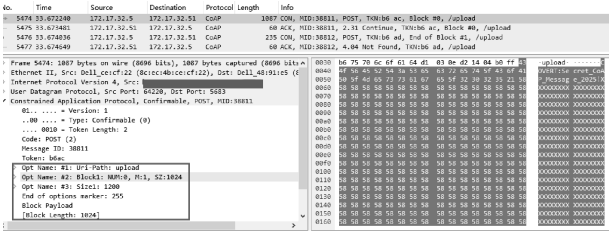


图 7　CoAP 协议 Block-wise 传输隐蔽信息

为了简化程序设计，发送端和接收端共享 32 bit 加密秘钥，数据块大小指数 $N$ 设置为 0（即式（2）中 $N$ 取 0），Block-wise 块传输每个块大小相同，设定为 $2^4$，即 16 bit 大小。Nonce 为 12 bit 的随机数，Plaintext 为编码后的隐蔽数据，AAD 采用 None 值。图 8 所示是采用 AES-GCM 认证加密算法，利用 Block-wise 块传输隐蔽信息的网络抓包。

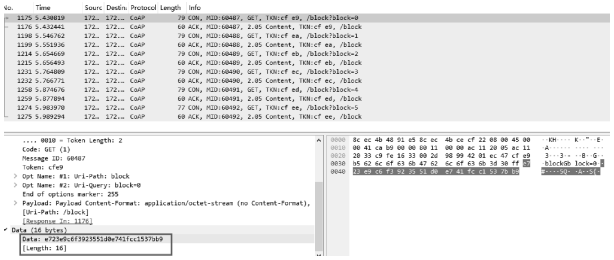


图 8　Block-wise 块隐藏加密消息

从图 8 可以看出，发送端发送带 Block 块编号的 GET 请求数据包。加密后的数据分块隐藏在 Payload 字段，每个数据包传输的 Payload 字段的大小为 16 B。如果传输数据量特别大，每个数据包 Payload 字段可以传输 1 024 B。服务器端接收到消息后根据块编号排序所有 Block 数据，最后通过 AES-GCM 算法进行解密即可还原原始消息。

图 9、图 10 所示分别是发送端利用 Block-wise 块发送数据和接收端解密还原消息的结果。

```
F:\1202\Covert channel>python3 coap_block_client2.py
[Client] 原始消息: 'This is CoAP Covert Channel Using Block To Transport Encrypt Data!'
[Client] 加密后长度: 94 字节
[Client] 分为 6 个 Block 发送
  Block 0 (编号=0): 发送成功, 响应=2.05 Content
  Block 1 (编号=1): 发送成功, 响应=2.05 Content
  Block 2 (编号=2): 发送成功, 响应=2.05 Content
  Block 3 (编号=3): 发送成功, 响应=2.05 Content
  Block 4 (编号=4): 发送成功, 响应=2.05 Content
  Block 5 (编号=5): 发送成功, 响应=2.05 Content
```

图 9　发送端发送加密数据

```
F:\1202\Covert channel>python3 coap_block_server2.py
=== CoAP Block-wise 隐蔽信道服务器启动 (UDP4, 172.17.32.161:5683) ===
[Server] 收到 Block 编号: 0, 数据长度: 16
[Server] 收到 Block 编号: 1, 数据长度: 16
[Server] 收到 Block 编号: 2, 数据长度: 16
[Server] 收到 Block 编号: 3, 数据长度: 16
[Server] 收到 Block 编号: 4, 数据长度: 16
[Server] 收到 Block 编号: 5, 数据长度: 14
[Server] 收齐全部 6 块, 总长度: 94
[Server] ✓ 成功解密! 明文: This is CoAP Covert Channel Using Block To Transport Encrypt Data!
```

图 10　接收端发送解密数据

从以上抓包数据和实验结果可以看出，改进后隐蔽通道对隐蔽流量进行加密后传输，提升了隐蔽性，同时，AES-GCM 算法提供完整性和认证的校验功能，利用 CoAP 协议 CON/ACK 确认机制防止数据丢包，保证了数据流的可靠性。

## 4　CoAP 协议安全防护建议

CoAP 协议作为物联网感知层核心通信协议，因轻

量化、低功耗的设计特性，在扩充协议功能的同时也带来安全隐患，攻击者可以利用冗余字段构建隐蔽通道。典型类型包括基于 Option 字段冗余、Token 字段伪装、报文 Payload 隐写的隐蔽通道，主要特征有：

（1）隐蔽性强。依托 CoAP 协议合法报文结构，将秘密数据嵌入非冗余字段，流量格式与正常通信高度一致，可规避传统防火墙、IDS 的规则化检测。

（2）适配性强。契合物联网终端低带宽、低算力场景，无需复杂加密或封装操作，传输成本低且易部署。

（3）加密适配性强。部分隐蔽通道利用加密机制，将隐写数据嵌入加密流量中，进一步提升检测难度，同时存在字段异常冗余等隐性特征。

针对上述 CoAP 协议隐蔽通道特征，结合协议特性，本文从三类技术方向给出安全防护建议：

（1）在协议异常检测层面，需基于 CoAP 协议规范构建基线模型，明确 Option 字段类型、长度、取值范围，详细分析 Token 字段和 Option 字段上下文是否存在相互联系。

（2）在深度报文解析（DPI）层面，需实现 CoAP 报文全字段精细化解析，重点排查非必要字段的冗余数据，对 Option 字段中的自定义选项、负载数据中的非标准格式内容进行深度校验，通过利用多编码自动检测探测工具（如 CyberChef）识别编码数据，利用网络流量解析工具（如 NetworkMiner）识别并解码 Base64 编码数据，组合协议前后数据包的 Payload 字段并利用 binwalk 等工具提取隐藏的编码数据块。

针对加密场景下的隐蔽通道，可从流量统计特征识别、协议行为深度校验以及基于上下文与场景的辅助识别三个方面开展加密隐蔽通道的检测。

（3）在机器学习驱动[12-13]的流量分析层面，需构建一套面向加密 CoAP 协议的多粒度、多维度特征提取体系，应从空间域、协议域和统计域三个维度协同采集特征。在空间域，采集单个 CoAP 报文长度、会话总字节数、ACK/CON/RST 消息比例、重传次数等反映异常通信模式。在协议域，聚焦于 Token、MsgID、Option 字段的取值分布、组合规律及复用频率。在统计域，引入 n-gram 字节序列频次、压缩比（如使用 LZ77 算法）、字节分布均匀度等无监督指标，捕捉载荷中隐藏的非自然随机性。在此基础上，采用集成分类算法（如随机森林、XGBoost）或核方法（如支持向量机 SVM）构建高鲁棒性的二分类或异常检测模型识别隐蔽通道。针对利用 Block2 选项分块传输机制，静态模型极易失败，引入增量学习机制，利用滑动时间窗口持续注入新样本，动态更新模型参数，适配新型 CoAP 隐蔽通道变体，持续优化模型泛化能力，实现对未知隐蔽通道的精准识别与阻断。

## 5 结论

本文利用物联网 CoAP 通信协议构建了三个不同的隐蔽通道，在实验环境下验证了该隐蔽通道能够穿透防火墙，躲避杀毒软件，成功实现隐蔽信息传输。分析了 CoAP 协议隐蔽通道特征，针对 CoAP 协议的可选字段存在数据隐写的安全隐患提出相应的安全防护建议。同时本文研究结果有助于理解资源受限环境下的隐蔽通信机制，并为构建下一代物联网系统的强健安全框架奠定了基础。该研究延伸到工业物联网、智慧城市等关键基础设施的网络安全防护，可为快速发展的物联网领域网络安全专业人员和研究人员提供有价值的参考。

**参考文献**


[1] 郭蕊，杜彦辉，芦天亮，等. 基于 CoAP 协议参数序列的隐蔽信道研究［J］. 计算机应用与软件，2021，38（8）：138－143.

[2] 邓雨欣，唐彰国，张健，等. 基于 MQTT 协议命令分组编码的隐蔽信道研究［J］. 计算机工程，2019，45（11）：138－143.

[3] 李凤华，李超洋，郭超，等. 泛在网络环境下隐蔽通道关键技术研究综述［J］. 通信学报，2022，43（4）：186－201.

[4] MILEVA A，VELINOV A，HARTMANN L. Comprehensive analysis of MQTT 5.0 susceptibility to network covert channels［J］. Computers & Security，2021，104. DOI：10.1016/j.cose.2021.102207.

[5] VELINOV A，MILEVA A，WENDZEL S，et al. Covert channels in the MQTT-based Internet of Things［J］. IEEE Access，2019，7：161899－161915.

[6] 常慧妍，扈红超，周大成，等. 基于应用程序的隐蔽信道研究综述［J/OL］. 计算机科学，1－22［2026－01－10］. http：//link. cnki. net/urlid/50.1075.TP.20250717.1544.020.

[7] 李彦峰，丁丽萍，吴敬征，等. 网络隐蔽信道关键技术研究综述［J］. 软件学报，2019，30（8）：2470－2490.

[8] 周正. 基于物联网的物理隔离隐蔽信道研究［D］. 合肥：中国科学技术大学，2019.

[9] 郭涛，赵自强. 网络隐蔽信道构建关键技术研究［J］. 网络安全技术与应用，2025（2）：3－5.

[10] 雷志群. 一种基于 AES-GCM 的数据完整性校验方法［J］. 计算机与数字工程，2016，44（11）：2229－2235.

[11] AHMD N，WEI L M，HAIROL JABBAR M. Advanced encryption standard with galois counter mode using field programmable

gate array [C]//International Conference on Green and Sustainable Computing (ICoGeS), 2017.
[12] CAVIGLIONE L. Trends and challenges in network covert channels countermeasures [J]. Applied Sciences, 2021, 11 (4). doi: 10.3390/app11041641.
[13] GUARASCIO M, ZUPPELLI M, CASSAVIA N. Detection of network covert channels in IoT ecosystems using machine learning [C]//ITASEC'22: Italian Conference on Cybersecurity, 2022.

(收稿日期: 2026-01-15)

作者简介:
姬国珍 (1984-), 男, 硕士, 高级工程师, 主要研究方向: 工业控制系统安全、协议分析。
康荣保 (1981-), 男, 博士, 研究员级高级工程师, 主要研究方向: 网络安全。
田学成 (1993-), 男, 硕士, 工程师, 主要研究方向: 电力系统安全。

## "工业互联网安全技术研究" 主题专栏征稿启事

工业互联网作为新一代信息技术与制造业深度融合的产物, 是推动产业数字化转型、实现经济高质量发展的关键基础设施。然而, 随着其广泛部署与深度应用, 网络攻击手段不断演进, 安全风险日益凸显。工业互联网安全不仅关乎生产系统的稳定运行, 更影响着关键基础设施的安全底线与数字经济的健康发展。为集中展示我国在工业互联网安全领域的最新理论研究成果、核心技术突破与创新应用实践, 推动构建自主、安全、可靠的工业互联网安全保障体系, 本刊拟在2026年第6期推出"工业互联网安全技术研究"主题专栏, 现面向国内外广大专家学者、科研人员及行业工程师公开征稿。

### 一、征文主题: 工业互联网安全技术研究

包括但不限于以下学术方向:

1. 工业互联网安全参考架构与标准体系;
2. 工控协议深度分析与安全加固;
3. 工业入侵检测与威胁感知;
4. 数据安全与隐私保护;
5. 工业智能体安全;
6. 5G+MEC环境下的工业安全;
7. 供应链安全 (第三方组件、软件库、开源工具的安全管控);
8. 人工智能/机器学习用于攻击检测与防御;
9. 区块链技术在工业数据完整性、溯源方面的应用。

### 二、投稿要求

1. 稿件请用word格式录入, 并套用本刊投稿模板。模板下载网址: http://files.chinaaet.com/files/Periodical/pcachina_Templates.doc

2. 投稿文章须未在其他期刊或者出版正式论文集的会议上刊登过, 且不在其他刊物或会议的审稿过程中, 不存在一稿多投现象。

3. 保证文章的合法性 (无抄袭、剽窃、侵权、虚假引用等不良学术行为), 且不违反相关法律法规, 不涉及国家、企业秘密, 稿件文责自负。

4. 论文要求观点鲜明、逻辑严谨、论据充分、方法合理, 字数在5000~8000字。

5. 请在官方投稿网站 (http://www.pcachina.com) 注册、投稿。注册后请投稿在"主题专栏"栏目, "稿件标题"请填"工业互联网+文章题目"。稿件经评审合格录用后, 在《网络安全与数据治理》2026年第6期 (正刊) 以主题专栏形式发表。

### 三、专栏主编

洪晟, 北京航空航天大学副教授, 博士生导师, 北京市安全学科带头人, 北京市科委技术专家, 北京航空航天大学"青年拔尖人才"。主持和参与973/863课题、国家重点研发课题、技术基础课题、国家自然科学基金等课题30余项; 担任国内外多个期刊和会议的编委, 发表论文70余篇, 其中SCI检索30余篇, 全球前1% ESI高被引论文1篇; 授权国家发明专利20项, 获国防科技进步一等奖1项, 北京市科学技术进步二等奖1项。

### 四、时间安排

截稿日期: 2026年4月30日
审稿反馈日期: 2026年5月15日
出版日期: 2026年6月15日

《网络安全与数据治理》编辑部
2026年2月

# 版权声明

凡《网络安全与数据治理》录用的文章，如作者没有关于汇编权、翻译权、印刷权及电子版的复制权、信息网络传播权与发行权等版权的特殊声明，即视作该文章署名作者同意将该文章的汇编权、翻译权、印刷权及电子版的复制权、信息网络传播权与发行权授予本刊，本刊有权授权本刊合作数据库、合作媒体等合作伙伴使用。同时，本刊支付的稿酬已包含上述使用的费用，特此声明。

《网络安全与数据治理》编辑部