---------------------------------------------------------------------

# 文档声明

## 合规性提示

本文档仅用于VPN技术原理及应用场景的知识科普，不涉及任何翻墙工具的推荐、使用指导或技术破解内容。根据中国法律法规，未经许可私自使用VPN访问境外网络属于违法行为，相关使用需严格遵守国家网络安全管理规定及企业合规要求。

## 版权与使用说明

本文档所有内容（包括技术原理、分类说明、应用场景等）均为作者原创整理，仅供个人学习、研究参考使用，禁止用于任何商业用途（如付费课程、商业出版物、盈利性内容分发等），禁止以任何形式（包括完整转载、节选、截图、二次编辑后转发等）在其他平台传播。任何获取、使用本文档的用户，均需完整保留作者信息及本声明内容，不得抹去或修改原创标识。

## 内容免责

本文档内容仅为技术知识梳理，不构成专业指导建议。因引用本文档内容导致的任何直接或间接后果（包括但不限于信息误差、合规风险、第三方投诉等），作者不承担法律责任。文档中提及的技术方案（如VPN协议、实现流程）仅作原理说明，具体使用需结合当地法律法规及企业内部规范。

## 内容完善说明

本文档对VPN技术的介绍为基础梳理，部分细节（如各方案的深层技术参数、复杂场景配置等）将在后续版本中补充完善，当前内容仅供初步参考。

---------------------------------------------------------------------

# VPN

了解VPN的作用前，需要分清楚“虚拟”、“私有”、“网络”等基础概念：

VPN是“虚拟的”，因为VPN是一种不用改变物理设备的技术。

VPN是“私人的”，因为VPN会加密连接。

VPN是“网络”，因为VPN能够在设备—VPN服务器—互联网上建立安全连接。

VPN在大陆境内使用需要向政府申请，任何私人使用VPN进行翻墙都是违法的。在绝大多数的西方国家内，VPN是完全合法的。VPN即使能够加密连接，但并不能通过隐藏IP来保护隐私，因为VPN除了设备和互联网上的目的主机外，还有VPN服务提供商能够直到整个连接【虽然VPN服务商不能够知道通信双方的密钥，但是通信双方的IP是肯定知道的（除了这里的三方以外，任何其他人都不能够知道）】。

## 1 VPN的分类

### （1）按使用场景分

**远程访问 VPN（Remote Access VPN）**

适用场景：员工出差、居家办公时，通过个人设备（电脑、手机）访问企业内网（如 OA 系统、文件服务器）。

原理：用户设备（客户端）通过互联网连接企业的 VPN 网关，认证通过后建立加密通道，相当于把个人设备 “接入” 企业内网。

典型技术：PPTP（早期，安全性低，已淘汰）、L2TP/IPsec（L2TP 负责封装，IPsec 负责加密，兼容性好）、SSL VPN（通过浏览器或轻量客户端接入，无需复杂配置，主流选择）。

**站点到站点 VPN（Site-to-Site VPN）**

适用场景：企业总部与分支机构（如北京总部和上海分公司）之间的局域网互联，让两地内网设备（如服务器、打印机）能直接通信。

原理：在两地的网关设备（如路由器、防火墙）之间建立 VPN 通道，两地内网的所有设备无需单独配置，通过网关自动转发数据。

典型技术：IPsec VPN（最常用，通过 IPsec 协议在网关间建立加密隧道）、MPLS VPN（依托运营商MPLS网络，安全性由运营商保障，适合对稳定性要求高的企业）。

**混合 VPN**

结合远程访问和站点到站点功能，例如企业既允许员工远程接入，又实现总部与分公司互联。

### （2）按技术实现分（核心是“隧道协议”）

**IPsec VPN**

底层协议：基于IP协议，通过“隧道模式”或“传输模式”封装数据。

隧道模式：对整个原始IP数据包（包括头部）加密并重新封装，适合站点到站点场景。

传输模式：仅加密数据包的“数据部分”，保留原始IP头部，适合两个主机直接通信。

优势：安全性强（支持加密、认证、防重放），标准化程度高，几乎所有网络设备都支持。

不足：配置较复杂，对网络带宽和延迟较敏感。

**SSL/TLS VPN**

底层协议：基于SSL/TLS（常用于HTTPS的加密协议），通过TCP端口（如 443）传输，可穿透大多数防火墙。

优势：无需安装专用客户端（部分场景支持浏览器直接访问），配置简单，适合远程访问（尤其是移动设备）。

不足：主要针对应用层数据（如网页、文件），对底层网络协议（如TCP/UDP）的支持较弱。

**MPLS VPN**

底层依赖：依托运营商的MPLS网络（非公共互联网），通过MPLS的LSP（标签交换路径）实现“逻辑隔离”，而非加密。

原理：运营商为企业分配独立的“虚拟路由转发（VRF）” 实例，不同企业的流量在 MPLS 网络中通过标签隔离，互不干扰。

优势：稳定性高（依托运营商专线级网络）、延迟低，适合对 QoS（服务质量）要求高的场景（如视频会议、ERP 系统）。

不足：需向运营商购买服务，成本较高；安全性依赖运营商网络隔离（非加密，适合信任运营商的场景）。

**其他小众类型**

PPTP/L2TP：早期协议，PPTP安全性差（加密弱），已基本淘汰；L2TP需配合IPsec使用（即 L2TP/IPsec），兼容性好但效率较低。

WireGuard：新兴轻量级协议，加密效率高、配置简单，逐渐替代传统IPsec，被Linux、Windows等系统原生支持。

## 2 VPN的实现

VPN（Virtual Private Network，虚拟专用网络）是一种通过公共网络（如互联网）建立“虚拟专用通道”的技术，能在不安全的公共网络中实现私密、安全的数据传输，让远程用户、分支机构像连接本地局域网一样访问企业内部资源。

VPN的本质是“封装+隔离+加密”的组合：

封装（隧道技术）：将私有网络的数据“包裹”再公共网络可传输的数据包中，通过公共网络传输。

隔离（私有基础）：通过地址规划、路由控制、逻辑区分等方法确保私有网络数据不会被公共网络或者其他非授权VPN的用户访问。

加密与认证：对封装的数据进行加密，并验证发送方省份和数据完整性，防止被窃取和篡改。

### （1）IPsec VPN（基于IP层的加密隧道，适合站点到站点、远程接入）

IPsec（IP Security）是一套在IP层实现加密和认证的协议族，通过“隧道封装”和“安全协议”实现私有性和保密性，是企业跨站点互联的主流选择。

实现流程（以“站点到站点”为例）：

**步骤 1**：建立逻辑隧道。两个私有网络（如总部和分支）的网关（路由器 / 防火墙）在互联网上协商建立“IPsec隧道”：

源网关将本地私有 IP 数据包（如192.168.1.0/24）封装到“公共IP数据包”中（外层IP为网关的公网地址）。

隧道通过“目标网关公网地址”定位，仅两端网关能识别和处理隧道内的私有数据（类似“专属快递通道”，只有收发双方能拆包）。

**步骤 2**：安全协议保障（核心：私有+保密）。IPsec通过AH（认证头）和ESP（封装安全载荷）协议实现安全，其中ESP 主流（同时支持加密和认证）：

加密（保密性）：对私有数据（或整个数据包）使用对称加密算法（如AES-256）加密，即使被截获，无密钥也无法解密。

认证（私有性+防伪装）：通过哈希算法（如SHA-256）对数据和发送方身份进行认证，确保数据来自授权网关（防止第三方伪装）。

数据完整性（防篡改）：通过校验值验证数据在传输中未被修改（如ESP的完整性校验字段）。

**步骤 3**：密钥管理（确保加密安全）

加密和认证需要密钥，IPsec通过IKE（互联网密钥交换）协议自动协商密钥：

双方先通过“预共享密钥”“数字证书”等方式验证身份（防止中间人攻击）。

动态生成临时会话密钥（避免长期使用同一密钥被破解），密钥仅在两端网关存储，不在公共网络传输。

### （2）SSL VPN（基于应用层的加密，适合远程用户接入）

SSL VPN基于SSL/TLS协议（与HTTPS的安全机制相同），无需在用户设备部署复杂客户端，适合员工远程访问企业内网（如通过浏览器或轻量客户端）。

实现流程（以远程访问企业OA为例）：

**步骤 1**：建立SSL/TLS连。用户通过浏览器访问企业SSL VPN网关（公网地址），网关与用户设备协商SSL/TLS会话：

网关出示数字证书（用户验证网关身份，防止钓鱼）。

双方协商加密算法（如AES）和会话密钥（通过非对称加密传输对称密钥，确保密钥安全）。

**步骤 2**：权限验证与资源隔离（私有性）。

用户输入账号密码（或二次认证，如动态口令），网关验证通过后，仅开放其权限内的资源（如仅允许访问OA系统，不允许访问财务服务器）。

企业内网资源使用私有IP（如10.0.0.0/8），用户设备通过“端口转发”、“Web 代理”等方式访问，数据被封装在SSL/TLS隧道中，公共网络无法直接访问内网IP（隔离性）。

**步骤 3**：加密传输（保密性）。用户与内网资源的所有数据（如 OA 登录信息、文件）均通过 SSL/TLS 加密：

数据在发送前用会话密钥加密，到达网关后解密。

SSL/TLS 的“握手阶段”确保密钥不被窃取，“记录协议”确保数据传输中加密且不可篡改。

### （3）MPLS VPN（基于运营商网络的私有链路，适合企业专线互联）

MPLS VPN由运营商提供，通过MPLS技术在公共骨干网中划分“逻辑私有网络”，适合多分支企业（如全国连锁公司）互联，私有性依赖网络隔离，保密性可额外叠加加密。

实现流程（以总部-分支互联为例）：

**步骤 1**：逻辑隔离（私有性核心）。运营商为企业分配独立的“VPN 实例”（类似 “虚拟路由器”）：

每个 VPN 实例有独立的路由表（仅包含该企业的分支地址，如总部 192.168.1.0、分支192.168.2.0）；

通过“路由目标（RT）”控制路由传播：仅携带相同RT的路由能进入同一VPN 实例（其他企业的路由无法接入）；

数据通过MPLS标签转发（标签仅运营商设备识别），公共网络无法解析标签对应的私有路由（隔离性）。

**步骤 2**：标签转发与可选加密（保密性）。

基础MPLS VPN的标签转发本身不加密（依赖运营商网络安全性）；

若需高保密，可在企业网关间叠加IPsec（即“MPLS+IPsec”）：企业数据先经IPsec加密，再进入MPLS隧道转发，双重保障（运营商看不到明文，公共网络无法破解）。

### （4）WireGuard VPN（轻量高效的现代加密隧道，适合个人及小型网络）

WireGuard是基于现代加密算法的轻量VPN技术，摒弃复杂协议栈，通过公私钥验证和高效加密实现私有性和保密性，启动快、配置简单，适合个人远程访问、小型网络互联。

实现流程（以“个人设备远程访问家庭设备”为例）：

**步骤 1**：预配置密钥与授权列表（私有性基础）。

服务端（家庭网关）和客户端（个人设备）各自生成“公私钥对”（私钥本地保存，公钥用于身份识别）。

双方预配置“授权列表”：服务端记录客户端公钥及允许访问的私有网段（如 192.168.3.0/24），客户端记录服务端公钥及公网地址。

效果：仅在授权列表中的设备可发起连接，非授权设备无法接入。

**步骤 2**：建立加密隧道（保密性核心）。

客户端访问家庭私有设备时，将私有数据包封装为“WireGuard数据包”（外层为双方公网IP，内层为私有IP）。

加密与认证：用服务端公钥加密数据（仅服务端私钥可解密），用客户端私钥生成数字签名（验证身份）。

加密算法：采用 ChaCha20（加密）和 Ed25519（签名），效率高于传统算法，防窃听、防伪装。

**步骤 3**：数据转发与私有访问（私有性落地）。

服务端接收后，用私钥解密并验证签名（确认来自授权客户端）。

将数据转发至家庭私有设备（如NAS），反向数据按相同流程加密回传。

隔离性：客户端仅能访问预配置的私有网段，不暴露其他网络，私有IP不被公网感知。

### 总结

|  |  |  |
| --- | --- | --- |
| **目标** | **核心技术** | **说明** |
| 私有性 | 隧道封装 | 将私有数据包裹在公共网络数据包中，公共网络仅传输 “封装后的外壳”，无法识别内部私有数据。 |
|  | 地址隔离 | 使用私有IP地址（如192.168.x.x），与公共IP隔离；通过VPN实例、路由控制限制访问范围。 |
|  | 身份认证 | 仅授权用户/设备（通过账号密码、证书、密钥）可接入VPN，拒绝非授权访问。 |
| 保密性 | 数据加密 | 对传输数据用对称加密（如AES），即使被截获，无密钥也无法解密。 |
|  | 密钥管理 | 通过IKE、SSL握手等动态协商密钥，密钥不在公共网络传输，且定期更新。 |
|  | 哈希校验 | 验证发送方身份（防伪装）和数据未被篡改（如哈希校验），确保接收的数据真实完整。 |

VPN的“私有性”本质是通过逻辑隔离（隧道、地址、路由）和身份控制，让数据在公共网络中“像在专用网络中一样仅被授权方访问”；“保密性”则通过加密算法、密钥管理、认证机制，确保数据在传输中不被窃取或篡改。不同VPN 技术（IPsec、SSL、MPLS等）的实现细节不同，但核心都是通过“隔离+加密”的组合，在公共网络上构建安全的私有通信链路。

**受限与篇幅，VPN的四种方案其实介绍的比较笼统，作者对此并不满意，作者会在之后进一步完善。**

## 3 VPN应用场景

**企业远程办公**

员工通过SSL VPN或IPsec VPN客户端，接入企业内网，访问办公系统、下载文件，体验与坐在公司工位相同。

**企业分支机构互联**

总部与分公司通过Site-to-Site IPsec VPN或MPLS VPN互联，实现两地内网设备（如服务器、打印机）的直接通信，例如上海分公司电脑可直接访问北京总部的数据库。

**跨境访问优化**

部分地区因网络限制无法访问特定服务（如国际学术数据库、海外办公系统），通过VPN将流量路由到目标地区的节点，实现合规访问（需注意当地法律法规）。

**保护公共网络通信**

在公共WiFi（如咖啡厅、机场）等不安全环境中，使用个人VPN加密上网数据，防止账号密码、支付信息被窃听。

**本文章归属 github 用户 WhatTheFuck-cyber**