

密码学原理

实验四：Web PKI 与 TLS

学号：2022113564

姓名：张哲恺

实验目的：通过搭建一个 HTTPS 网站并分析 Web PKI 与 TLS 协议来理解密码学的现实应用。

实验内容：

1. 建立一个部署数字证书的网站

要求：注册一个域名，建立一个 Web 网站，生成该网站的数字证书并部署，通过 HTTPS 访问该网站。域名可免费申请。Web 服务器推荐购买云服务。证书可以通过自建 CA 颁发（需在浏览器部署 CA 信任锚），或者向第三方 CA（Let's Encrypt、GoDaddy 等）申请。

首先是在阿里云上注册域名然后进行域名解析。

修改记录 ×

● 解析记录变更后，可能不会立即生效。因为各地网络运营商 dns 存在缓存，在缓存未到期时，是不会向云解析 DNS 请求最新的解析记录，而是直接将之前缓存的解析结果返回给访问者，所以需要等待运营商刷新本地缓存后，解析才会实际生效。解析生效时间主要取决于运营商 DNS 缓存的 TTL 到期时间，预计最快 10-30 分钟左右生效。如进行过 DNS 服务器名称修改，则一般需要 24-48 小时左右生效。 [了解更多](#)

记录类型 [查看帮助文档](#)

CNAME- 将域名指向另外一个域名

* 主机记录 [?](#)

zzk .chenrling.cn [?](#)

解析请求来源 [?](#)

指定域名访问者所在的地区和使用的运营商网络。

默认 - 必填！未匹配到智能解析线路时，返回【默认】线路设置结果

[升级至企业版DNS](#)，支持按更精细线路（省份、国家）请求来源返回不同记录值。

* 记录值 [?](#)

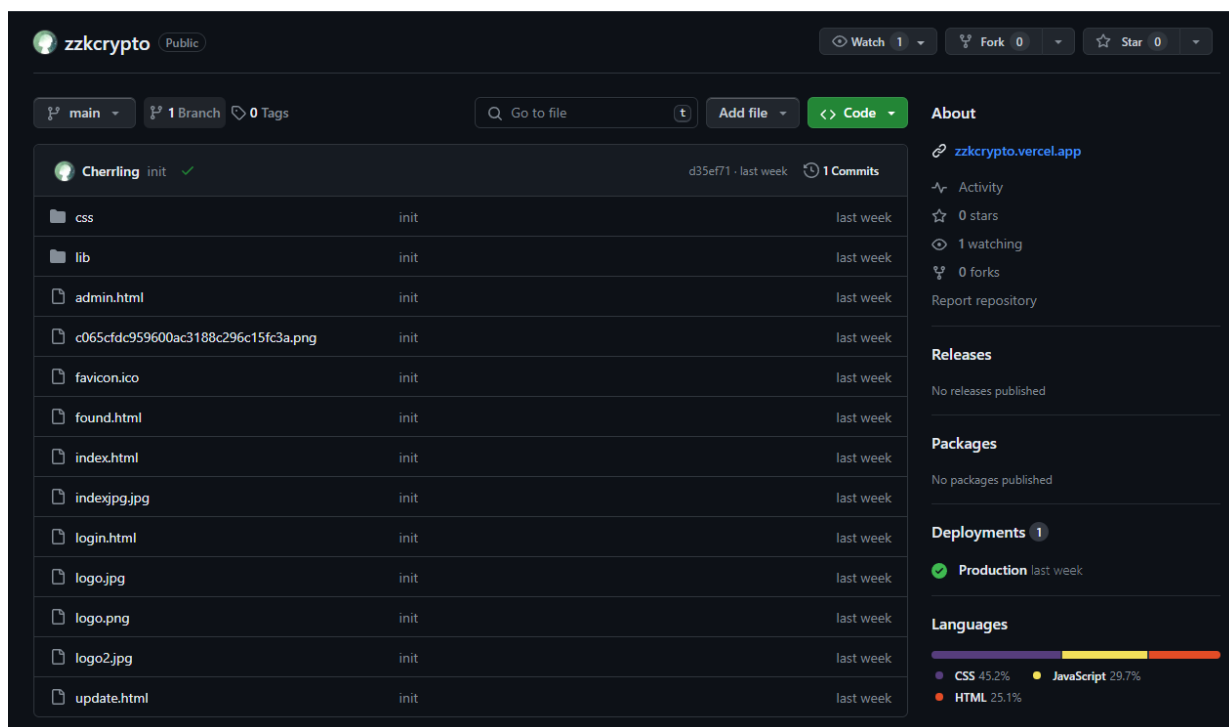
cname.vercel-dns.com

* TTL [?](#)

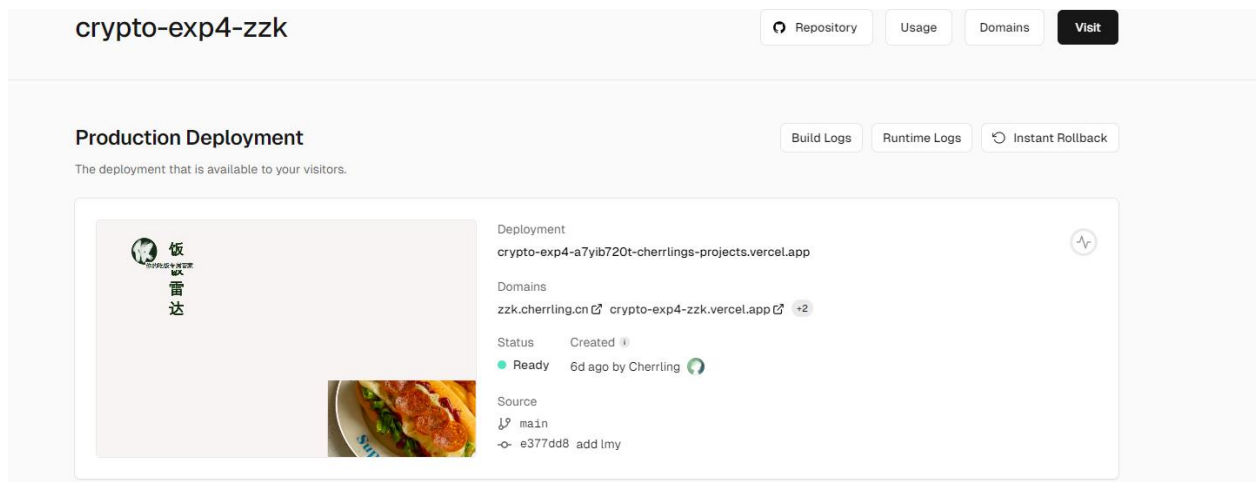
10 分钟

[升级至企业版DNS](#)，TTL最小可设置1秒。

使用 vercel 协助部署网站，首先把自己的网页文件上传到 GitHub 仓库。

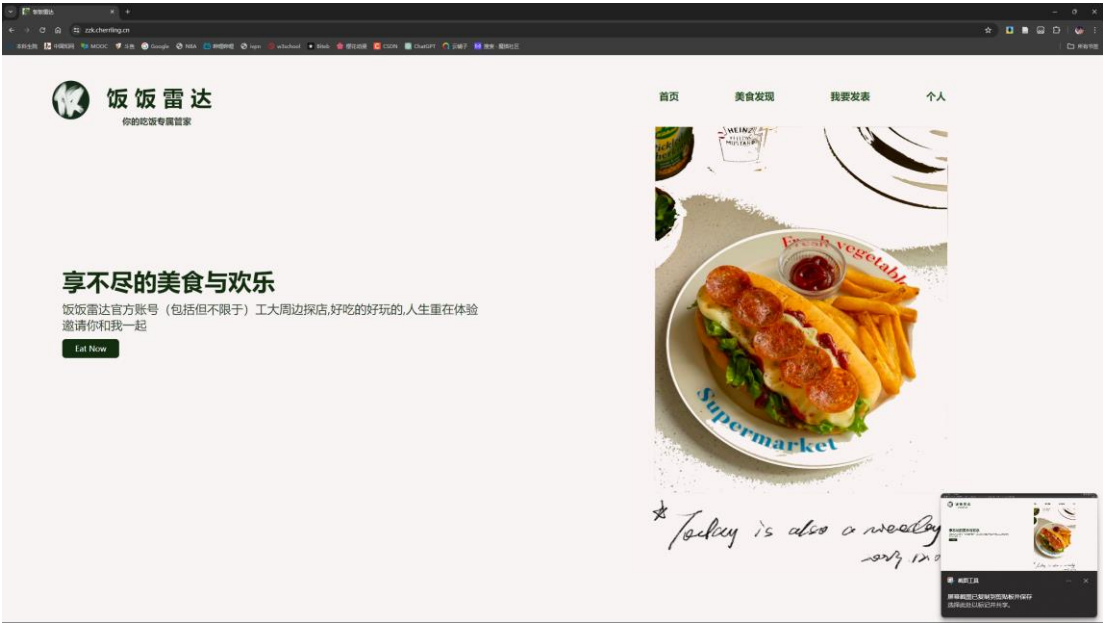


然后在 vercel 上导入刚刚的 GitHub 仓库进行网页部署。



然后将我们刚刚注册的域名与网页绑定即可。Vercel 会自动给我们颁发证书。

我的网页界面：



证书：

证书查看者: zzk.cherling.cn

基本信息(G)

详细信息(D)

颁发对象

公用名 (CN)	zzk.cherling.cn
组织 (O)	<未包含在证书中>
组织单位 (OU)	<未包含在证书中>

颁发者

公用名 (CN)	R3
组织 (O)	Let's Encrypt
组织单位 (OU)	<未包含在证书中>

有效期

颁发日期	2024年4月13日 星期六 16:19:53
截止日期	2024年7月12日 星期五 16:19:52

SHA-256 指纹

证书	60334277f85e1b0cebf20ccd4fa2ce1263848dfc0519a43c30d9d3860337c1e2
公钥	e75c190f436a7d92d04ac37411415fae97482f014e27dc1a013c124589a4b338

2. TLS 协议的密码学要素分析

要求：参考 TLS1.3 协议，通过浏览器自带功能和网络抓包等协议分析方法，详细展示其中的密码学要素信息，包括所涉及的密钥协商协议、数字签名(证书)、

非对称加密和对称加密方案。

访问支持 TLS1.3 协议的网站，使用浏览器自带功能控制台查看其密码学相关信息：



使用 wireshark 抓包，过滤 ip 地址。



抓包结果：

10795	150.721531	172.20.113.58	104.16.86.20	TLSv1.3	452 Client Hello (SNI=cdn.jsdelivr.net)
10808	151.059775	104.16.86.20	172.20.113.58	TLSv1.3	1414 Server Hello, Change Cipher Spec
10810	151.059775	104.16.86.20	172.20.113.58	TLSv1.3	629 Encrypted Extensions, Compressed Certificate, Certificate Verify, Finished
10812	151.061126	172.20.113.58	104.16.86.20	TLSv1.3	118 Change Cipher Spec, Finished

对握手的各个阶段进行分析

Client Hello: 客户端发送 Hello，传输支持的密码学套件和公钥。

Random 随机数

Session ID 会话 ID，

Cipher Suites 客户端发送自己支持的密码学套件

客户端指定使用 ECDHE 密钥交换模式：

```
Extension: psk_key_exchange_modes (len=2)
  Type: psk_key_exchange_modes (45)
  Length: 2
  PSK Key Exchange Modes Length: 1
  PSK Key Exchange Mode: PSK with (EC)DHE key establishment (psk_dhe_ke) (1)
```

客户端提供的签名算法：

```
▼ Extension: signature_algorithms (len=18)
  Type: signature_algorithms (13)
  Length: 18
  Signature Hash Algorithms Length: 16
  ▼ Signature Hash Algorithms (8 algorithms)
    ▶ Signature Algorithm: ecdsa_secp256r1_sha256 (0x0403)
    ▶ Signature Algorithm: rsa_pss_rsae_sha256 (0x0804)
    ▶ Signature Algorithm: rsa_pkcs1_sha256 (0x0401)
    ▶ Signature Algorithm: ecdsa_secp384r1_sha384 (0x0503)
    ▶ Signature Algorithm: rsa_pss_rsae_sha384 (0x0805)
    ▶ Signature Algorithm: rsa_pkcs1_sha384 (0x0501)
    ▶ Signature Algorithm: rsa_pss_rsae_sha512 (0x0806)
    ▶ Signature Algorithm: rsa_pkcs1_sha512 (0x0601)
```

客户端支持的椭圆曲线类型

```
Extension: supported_groups (len=12)
  Type: supported_groups (10)
  Length: 12
  Supported Groups List Length: 10
  ▼ Supported Groups (5 groups)
    Supported Group: Reserved (GREASE) (0x0a0a)
    Supported Group: X25519Kyber768Draft00 (0x6399)
    Supported Group: x25519 (0x001d)
    Supported Group: secp256r1 (0x0017)
    Supported Group: secp384r1 (0x0018)
```

客户端发送共享密钥数据，包括使用的椭圆曲线和对应的公钥。服务器收到后可以根据接受的 ECDHE 和客户端的公钥生成自己的公钥和对称密钥

```
▼ Extension: key_share (len=1263) X25519Kyber768Draft00, x25519
  Type: key_share (51)
  Length: 1263
  ▼ Key Share extension
    Client Key Share Length: 1261
    ▶ Key Share Entry: Group: Reserved (GREASE), Key Exchange length: 1
    ▶ Key Share Entry: Group: X25519Kyber768Draft00, Key Exchange length: 1216
    ▶ Key Share Entry: Group: x25519, Key Exchange length: 32
```

全部抓包结果：

```
▼ Frame 10795: 452 bytes on wire (3616 bits), 452 bytes captured (3616 bits) on interface \Device\NPF...
  Ethernet II, Src: Intel_34:7f:1f (f4:26:79:34:7f:1f), Dst: JuniperNetwo_d2:ff:c2 (44:ec:cc:e2:ff:c2)
  Internet Protocol Version 4, Src: 172.20.113.58, Dst: 104.16.86.20
  Transmission Control Protocol, Src Port: 6761, Dst Port: 443, Seq: 1361, Ack: 1, Len: 398
  [2 Reassembled TCP Segments (1758 bytes): #10794(1360), #10795(398)]
  ▼ Transport Layer Security
    ▼ TLSv1.3 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 1753
      ▼ Handshake Protocol: Client Hello
        Handshake Type: Client Hello (1)
        Length: 1749
        Version: TLS 1.2 (0x0303)
        Random: b0f6fdc7b8d0aac0b9d98269549dd4894b61075b553c6fa3dfa3a7ea88e2789
        Session ID Length: 32
        Session ID: ec1e2795e77110e064b2e354859f212828ca01939757728dab83ff59d66481df
        Cipher Suites Length: 32
        ▼ Cipher Suites (16 suites)
          Cipher Suite: Reserved (GREASE) (0x7a7a)
          Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
          Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
          Cipher Suite: TLS_CHACHA20_POLY1305_SHA256 (0x1303)
          Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
          Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
          Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc030)
          Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc03d)
          Cipher Suite: TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xcca9)
          Cipher Suite: TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xc0cb)
          Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)
          Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)
          Cipher Suite: TLS_RSA_WITH_AES_128_GCM_SHA256 (0x009c)
          Cipher Suite: TLS_RSA_WITH_AES_256_GCM_SHA384 (0x009d)
          Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
          Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
        Compression Methods Length: 1
        Compression Methods (1 method)
        Extensions Length: 1644
        ▶ Extension: Reserved (GREASE) (len=0)
        ▶ Extension: encrypted_client_hello (len=218)
        ▶ Extension: supported_versions (len=7) TLS 1.3, TLS 1.2
        ▶ Extension: application_settings (len=5)
        ▶ Extension: ec_point_formats (len=2)
        ▶ Extension: extended_master_secret (len=0)
        ▶ Extension: session_ticket (len=0)
        ▶ Extension: compress_certificate (len=3)
        ▶ Extension: supported_groups (len=12)
        ▶ Extension: key_share (len=1263) X25519Kyber768Draft00, x25519
        ▶ Extension: psk_key_exchange_modes (len=2)
        ▶ Extension: signature_algorithms (len=18)
        ▶ Extension: renegotiation_info (len=1)
        ▶ Extension: signed_certificate_timestamp (len=0)
        ▶ Extension: application_layer_protocol_negotiation (len=14)
        ▶ Extension: status_request (len=5)
        ▶ Extension: server_name (len=21) name=cdn.jsdelivr.net
        ▶ Extension: Reserved (GREASE) (len=1)
        [JA4: t13d1516h2_8daaf6152771_02713d6af862]
        [JA4_r: t13d1516h2_002f,0035,009c,009d,1301,1302,1303,c013,c014,c02b,c02c,c02f,c030,cca8,c0cb]
        [JA3 Fullstring: 771,4865-4866-4867-49195-49199-49196-49200-52393-52392-49171-49172-156-157]
        [JA3: c58b900e8b4ef1789e6ab169245764e]
```

Sever Hello: 服务器根据客户端的加密套件、公钥算出自己公钥和私钥。

服务器选择 AES_GCM 作为对称加密算法

```
Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
```

key_share 拓展，服务器把自己的公钥也发送给客户端，选择了 x25519 椭圆曲线

```
Extension: key_share (len=1124) X25519Kyber768Draft00
  Type: key_share (51)
  Length: 1124
  Key Share extension
    Key Share Entry: Group: X25519Kyber768Draft00, Key Exchange length: 1120
    Group: X25519Kyber768Draft00 (25497)
    Key Exchange Length: 1120
    Key Exchange [truncated]: 75ae6965c580362a68760e9d8c96f27369ee65717e3ba88af37149db
```

服务器告诉客户端改变加密方式，使用对称密钥加密。

```
TLSv1.3 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
  Content Type: Change Cipher Spec (20)
  Version: TLS 1.2 (0x0303)
  Length: 1
  Change Cipher Spec Message
```

全部抓包结果:

```
Frame 10808: 1414 bytes on wire (11312 bits), 1414 bytes captured (11312 bits) on interface \Device
Ethernet II, Src: JuniperNetwo_d2:ff:c2 (44:ec:ce:d2:ff:c2), Dst: Intel_34:7f:f1 (f4:26:79:34:7f:f1)
Internet Protocol Version 4, Src: 104.16.86.20, Dst: 172.20.113.58
Transmission Control Protocol, Src Port: 443, Dst Port: 6761, Seq: 1, Ack: 1759, Len: 1360
Transport Layer Security
  TLSv1.3 Record Layer: Handshake Protocol: Server Hello
    Content Type: Handshake (22)
    Version: TLS 1.2 (0x0303)
    Length: 1210
    Handshake Protocol: Server Hello
      Handshake Type: Server Hello (2)
      Length: 1206
      Version: TLS 1.2 (0x0303)
      Random: d25fb92d2e5d2754bdaabd815d4608b9cb7f7feb4e268820db167c2324928180
      Session ID Length: 32
      Session ID: ec1e2795e77110e064b2e354859f212828ca01939757728dab83ff59d66401df
      Cipher Suite: TLS_AES_128_GCM_SHA256 (0x1301)
      Compression Method: null (0)
      Extensions Length: 1134
      Extension: key_share (len=1124) X25519Kyber768Draft00
        Type: key_share (51)
        Length: 1124
        Key Share extension
          Key Share Entry: Group: X25519Kyber768Draft00, Key Exchange length: 1120
      Extension: supported_versions (len=2) TLS 1.3
        Type: supported_versions (43)
        Length: 2
        Supported Version: TLS 1.3 (0x0304)
        [JA3S Fullstring: 771,4865,51-43]
        [JA3S: eb1d94daa7e0344597e756a1fb6e7054]
      TLSv1.3 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
        Content Type: Change Cipher Spec (20)
        Version: TLS 1.2 (0x0303)
        Length: 1
        Change Cipher Spec Message
      TLS segment data (139 bytes)
```

Certificate: 为了确认身份，服务器把证书发给客户端，客户端再用第三方权威认证中心的公钥解析证书。

可以发现服务器发送的是 Compressed Certificate，发送压缩过后的证书能有效减少握手时的带宽消耗。

```
Handshake Protocol: Compressed Certificate
Handshake Type: Compressed Certificate (25)
Length: 1838
Algorithm: brotli (2)
Uncompressed Length: 2605
Length: 1830
  Compressed Certificate Message [truncated]: 81605100203f4fdbfcfee38238a6d56bc23cc04238250b150c0b928d886eb629db828
    Certificate Request Context Length: 0
    Certificates Length: 2601
    Certificates (2601 bytes)
      Certificate Length: 1327
      Certificate [truncated]: 3082052b308204d0a0030201020210072dbb27d467bf66ed70dd7ba132c622300a06082a8648ce3d040
      Extensions Length: 287
      Extension: status_request (len=283)
      Certificate Length: 977
      Certificate [truncated]: 308203cd308202b5a00302010202100a3787645e5fb48c224efd1bed140c3c300d06092a864886f70d0
      Extensions Length: 0
```

每个证书里还包含了证书的颁发者 issuer，有效期 validity，颁发给谁的 subject，服务器的公钥 subjectPublicKeyInfo

```
signedCertificate
  version: v3 (2)
  serialNumber: 0x072dbb27d467bf66ed70dd7ba132c622
  signature (ecdsa-with-SHA256)
  issuer: rdnSequence (0)
    rdnSequence: 3 items (id-at-commonName=Cloudflare Inc ECC CA-3,id-at-organizationName=Cloudfla
      RDNSequence item: 1 item (id-at-countryName=US)
      RDNSequence item: 1 item (id-at-organizationName=Cloudflare, Inc.)
      RDNSequence item: 1 item (id-at-commonName=Cloudflare Inc ECC CA-3)
  validity
  subject: rdnSequence (0)
    rdnSequence: 5 items (id-at-commonName=sni.cloudflaressl.com,id-at-organizationName=Cloudflar
  subjectPublicKeyInfo
    algorithm (id-ecPublicKey)
    Padding: 0
    subjectPublicKey: 04320d64c92d88e7ecd97435893195fd686f73a08d8c9b554286df5fe914bf9944b6a774695
```

Certificate verify: 客户端发送证书验证，使用 ecdsa_secp256r1 椭圆曲线来做认证，确保证书没有被篡改，hash 值是用 SHA256 计算出来的。

```
Handshake Protocol: Certificate Verify
Handshake Type: Certificate Verify (15)
Length: 76
  Signature Algorithm: ecdsa_secp256r1_sha256 (0x0403)
    Signature Hash Algorithm Hash: SHA256 (4)
    Signature Hash Algorithm Signature: ECDSA (3)
    Signature length: 72
    Signature: 3046022100f41e2b337efe528be7d56c832ae9f3120d14431b74b4028430f0591b99f47142022100b0135483d3cf3a208f646aee1178980b5
```

Finished: 这是身份验证阶段的最后一条消息，也是第一个使用协商的算法进行加密和防篡改保护的信息，Verify Data 使用 HMAC 计算得来的，包含 finished_key 和握手消息的 hash。

```
Handshake Protocol: Finished
Handshake Type: Finished (20)
Length: 32
Verify Data
```

TLS1.3 握手全过程：

