

## ET 系列动态令牌说明手册



## ET 系列动态令牌简介

坚石 ET 系列 OTP 动态令牌是一个内嵌特殊运算芯片的身份认证产品，结合了最高的移动性与安全性。令牌根据时间变化或者事件变化，动态随机产生一个新的 6 位或 8 位 OTP 密码。客户端无需安装任何软件，即可实现高强度身份认证需求，认证算法完全符合国际 OATH 组织标准。坚石 ET 系列 OTP 动态令牌因其简单、易用、小巧便携，被广泛应用于网络游戏、财务软件系统、ERP 软件系统、政府电子政务系统、军队系统、VPN 虚拟专网系统、金融系统等对密码安全要求较高的软件系统。

### ET 系列动态令牌的优势：

- ✓ 最简单易用的 OTP 产品，高效灵活嵌入到任何身份认证系统中
- ✓ ET z201 完全符合国际 OATH 标准组织 TOTP 算法
- ✓ ET z100 完全符合国际 OATH 标准组织 HOTP 算法
- ✓ 按键产生 6 位或 8 位 OTP 动态口令，口令一次有效
- ✓ 零学习使用，用户端无需安装任何硬件驱动或是中间软件
- ✓ OTP 硬件与计算机脱离，切断非法程序攻击入口
- ✓ 160 位密钥长度
- ✓ 防深水浸泡及煮沸高温，防水设计
- ✓ 防汽车碾压，抗压设计
- ✓ 防高层摔落，抗震设计
- ✓ 防静电、电磁辐射
- ✓ 电池寿命 3~5 年
- ✓ 按键寿命 10 万次
- ✓ 支持标准 RADIUS 协议
- ✓ 支持 Oracle、SQL Server、Postgre SQL、My SQL、Access 等数据库
- ✓ 认证服务器支持各种操作系统：Windows 全系列、Linux 全系列、Unix、Solaris、FreeBSD

### 令牌密钥：

- 令牌密钥是存储在硬件令牌中的关键信息，OTP 动态令牌硬件保证不能被非法获取
- 每个 OTP 动态令牌硬件中的密钥是唯一的，互不相同
- 令牌密钥也要存储在数据库中，用于服务器端对客户端传来的 OTP 进行认证。
- 符合国际 OATH 组织的规范标准，长度 160 位

# ET 系列动态令牌接口说明

使用 ET 系列动态令牌提供的认证接口可以灵活快速的将 OTP 解成到您的身份认证系统中。接口调用优点：

- ✓ 集成高效快速，不需要阅读繁琐的说明文档
- ✓ 集成方便灵活，不需改变现有认证体系
- ✓ OTP 数据库由开发商根据实际情况自定义
- ✓ 根据需求任意增加额外安全功能

## 1、认证接口

认证接口用于对 OTP 动态令牌硬件产生的 6 位或 8 位动态口令进行认证。

```
ET_CheckPwz201 (
    char *authkey,      //令牌密钥
                        //如: 1l598C28E54BF3C58EC35E830655A139861407519FCBC92140
    uint64_t t,          //当前 UTC 时间，即取自 1970 年 1 月 1 日到当前的一个数值
    uint64_t t0,         //给 0
    unsigned int x,      //给 60
    int drift,           //漂移值，上次调用成功后返回值（可以从数据库中取），第一次给 0
    int authwnd,         //认证窗口，默认 20，根据实际情况可以调整
    uint64_t lastsucc,   //成功值，上次调用成功后返回值（可以从数据库中取），第一次给 0
    const char *otp,     //OTP 口令，硬件按出的 6 位或 8 位 OTP
    int otpen,           //OTP 口令长度，给 6 或 8
    uint64_t *currsucc,  //返回成功值
    int *currdft         //返回漂移值
)
```

说明：

- authkey: 密钥为 50 个字符（前两个字符为小写的 1l（不是数字 1）+十六进制字符）
- 漂移值和成功值每次都要传入上一次调用（认证或者同步）成功后返回的值，第一次认证时给 0。因此应该每次调用（认证或者同步）成功后，将返回的 currsucc 和 currdft 存入到数据库中，下次调用认证接口时 drift 和 lastsucc 要从数据库中取出。**特别注意：只有调用（认证或者同步）成功时才将 currsucc 和 currdft 进行保存，供下次调用。调用（认证或者同步）失败时不必保存这两个值。**
- authwnd: 认证窗口。认证窗口是一个范围，即服务器端在这个范围内认证 OTP 是否有

效，由于计算机时间与 OTP 动态令牌硬件中的时间肯定有误差，因此认证窗口的存在是必要的。20 表示在计算机时间的前后各 10 分钟范围内进行认证。如果 OTP 动态令牌硬件中的时间比调用本接口的计算机时间快或慢 10 分钟内，那么认证就可以通过。如果偏差过大，认证就会失败。将认证窗口的值调大，可以增加认证成功的几率，但不宜超过 120。

## 2、同步接口

同步接口用于对认证不通过的令牌进行同步操作。当计算机时间与 OTP 动态令牌硬件中的时间偏差过大，超过认证窗口 `authwnd` 范围时，就会认证失败，这时需要进行同步操作。同步操作后，OTP 动态令牌可以继续使用认证接口进行认证。

```
ET_Syncz201 (
    char *authkey,      //令牌密钥
    uint64_t t,         //当前 UTC 时间，即取自 1970 年 1 月 1 日到当前的一个数值
    uint64_t t0,        //给 0
    unsigned int x,      //给 60
    int drift,          //漂移值，上次调用成功后返回值（可以从数据库中取），第一次给 0
    int syncwnd,        //同步窗口，默认 40，根据实际情况可以调整
    uint64_t lastsucc,   //成功值，上次调用成功后返回值（可以从数据库中取），第一次给 0
    const char *otp1,    //第一个 OTP 口令，硬件按出的 6 位或 8 位 OTP
    int otp1len,        //OTP 口令长度，给 6 或 8
    const char *otp2,    //第二个 OTP 口令，硬件按出的 6 位或 8 位 OTP
    int otp2len,        //OTP 口令长度，给 6 或 8
    uint64_t *currsucc,  //返回成功值
    int *currdft        //返回漂移值
)
```

说明：

- 同步时要输入两个连续的 6 位或 8 位 OTP 动态口令。
- 同步窗口与认证窗口的概念一样。由于同步只有在认证不通过时才调用，不是经常调用，因此可以将同步窗口设置的大一些，尽量保证 OTP 动态令牌硬件的时间能够在这个偏差范围内。
- 同步成功后注意要将返回的 `currsucc` 和 `currdft` 进行保存，供下回认证或者同步调用。
- 详细说明：如硬件中的时间是 8:00，而服务器（调用接口的计算机）的标准时间为 9:00，认证接口中 `authwnd` 参数默认为 20，即 9:00 的前后各 10 分钟，那么硬件中时间的范围在 8:50-9:10 时，硬件产生的 OTP 是可以认证通过的，但硬件中为 8:00，验证就不能通过。这时可以通过 2 个办法来解决这个问题，一种是调整认证接口中 `authwnd` 参数，设为 120，那么就是 9:00 的前后各 1 小时范围内，即 8:00-10:00，硬件中为 8:00，在这个

范围，可以验证通过。这种方法增加了服务器运算的范围，加重了负荷，因此不建议使用。另外一个解决办法就是同步运算，同步接口中窗口参数 `syncwnd` 可以设成 120，同步成功后将调整值 `drift` 和 `succ` 存入到数据库，供认证接口调用。认证接口再次调用时传入的是同步成功后的调整值，就能够弥补动态令牌硬件中时间和服务器时间的偏差了。由于同步接口只是在时间偏差过大，造成认证不通过才调用，被调用的概率很小，不会增加服务器计算的负荷，建议使用这种方法。

### 3、开发指南

- (1) 在您的数据库中增加一张用于存储 OTP 动态令牌信息的数据表。里面至少存储以下字段：“令牌号”（背面条形码）、“密钥”（`authkey`）、“成功值”（`currsucc`）、“漂移值”（`currdft`）。其中令牌号和密钥都可以用字符串形式，成功值和漂移值接口中要用到 `uint64` 和 `int` 类型。
- (2) 在系统的用户表中增加一个存“令牌号”的字段，存储与用户绑定的令牌号。

用户在登录时，输入用户名和 OTP 传给服务器端。服务器通过用户名到用户表中得到“令牌号”，再通过这个“令牌号”到令牌表中得到“密钥”，“成功值”和“漂移值”，带入到接口中进行认证或同步，认证或同步成功后将返回的值写回数据库中保存。认证或同步失败时，不要将这两个值写回数据库。