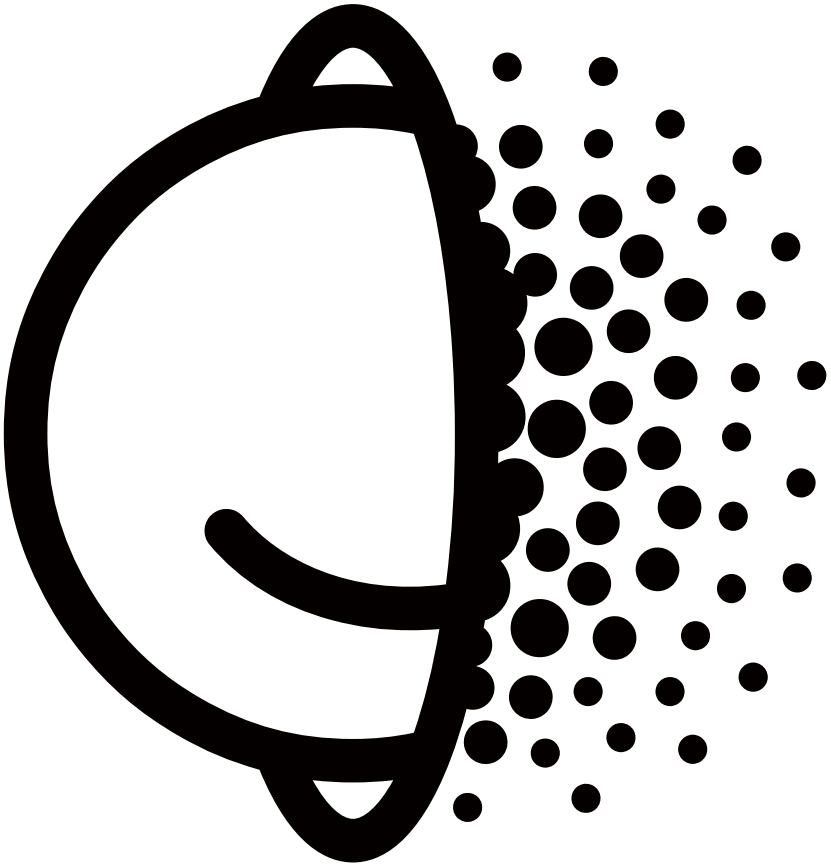


秘迹安全白皮书

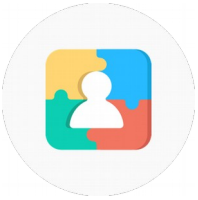
熵增（北京）网络科技有限公司

v1.0.0 (2019 年 1 月 25 日）

关键安全特性

端到端加密

保险箱数据以 AES-256 加密保存，加密密钥全世界只有你知道，这个密钥只存在你的客户端，秘迹的工程师也无法解密你的数据文件。



没有账号体系

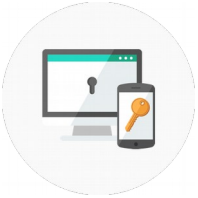
无需绑定手机、邮箱、用户名，杜绝个人信息泄漏，大大减少攻击面，完全匿名使用。

只需一个强密钥

使用和比特币钱包助记词一样的算法，数据通过 12 个汉字或单词的恢复密钥来解密。我们的密钥暴力破解需要 8.8 亿亿年，比用密码保护的账户更安全。

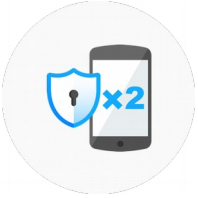
通过悄悄话安全分享密码

通过悄悄话功能在微信等渠道以加密方式分享密码等敏感数据，确保密码在秘迹之外也无法被轻易截获。



密码不离开手机登录电脑网站

通过跨屏登录，无需在电脑浏览器输入密码即可登录网站，既便捷又安全。



双层加密

秘迹客户端和秘迹服务器之间传输的保险箱数据都由用户密钥加密，传输通道又进行了 TLS 加密。两次加密，更加安全。

内容目录

设计原则

默认隐私

- 无注册流程，使用服务不需要提供手机号、邮箱、微信等任何个人可识别信息（Personally Identifiable Information）。
- 所有客户端程序只请求应用正常工作需要的最小的系统权限。

以客户端为核心

- 所有用户明文数据，包括密码、密钥、联系人、对话消息等，不离开客户端内存、不落地到本地或远程的任何持久存储介质。
- 用户之间加密通信内容只有用户指定的参与方才能解密，服务器只提供证书颁发、密文存储等功能。

不信任服务器

- 包括我们自己的服务器。防止我们的服务器或者系统管理员能看到用户数据的原则贯穿始终。
- 即使服务器被黑了，客户数据的安全也不受影响。

秘迹保险箱

险箱的定位是个人最私密的信息存储工具。它里面保存的信息，在主人主动分享给其他人或者其他服务之前，世界上都不应该知道这个信息的存在，更不用说信息的内容了。

恢复密钥

传统的互联网服务，包括密码管理器，都需要提供手机号、邮箱、用户名等个人可识别信息来注册、认证、授权账户。当某个网站的账号数据泄漏时，黑客会借助此数据在其它网站进行更有目标、更有效的攻击。

秘迹应用不要求用户提供个人可识别的身份信息，这样使用秘迹时是一个匿名的身份，秘迹用户的“身份”无法和其它任何互联网服务的账号体系关联起来。黑客无法通过其它网站泄漏的信息帮助攻击秘迹，减小了攻击面，让秘迹更安全。

绝大多数的互联网服务都用一组用户名和密码来认证用户。密码一般都在字母、数字、标点符号等可见字符里面选择，字符集大小受限。因为人性，大部分人往往会选择较短的、容易记忆，但是熵值低、安全性较差的密码。此外，用户经常在不同网站使用相同的密码，撞库攻击非常有效。由于以上原因，密码的安全性不够强。

秘迹应用不允许用户设置密码，而是使用一个在初次使用时随机生成的 128 位（Bit）数作为密钥。这个密钥的强度足够安全，远超绝大多数密码的强度，用目前最强的超级计算机暴力破解需要 8.8 亿亿年[1]。

为了方便记下密钥，我们使用和比特币 HD 钱包一样的方案 BIP39[2]来生成 12 个字的恢复密钥（Recovery Key）。使用秘迹保险箱安全保存数据，唯一需要记住的就是这十二个字的恢复密钥。保险箱数据在备份到服务器端时，都是本地加密后上传，加密的密钥只有用户掌握，服务器和秘迹的工程师无法看到保险箱明文。

保险箱内容加密方式

保险箱使用文件格式 KDBX 4。KDBX 文件使用 AES 加密算法，密钥长度 256 位，使用 CBC 模式（PKCS5Padding）。

认证方式

传统静态数据保护系统把账号认证和数据存储整合在了一起，会导致如下问题：

- 服务端可以冒充任意账号，进入任意账号，获得相应数据。
- 服务端可以封禁任意账号，账号主人即刻失去与所有数据。

1 How secure is AES against brute force attacks?, https://www.eetimes.com/document.asp?doc_id=1279619
2 Mnemonic code for generating deterministic keys, https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki

- 服务端往往要求用户绑定手机、邮箱、还有其他第三方认证服务来认证用户，从而导致服务端一旦被黑，大量用户资料就会泄漏。

与之相对应我们的服务端：

- 没有任何用户账号。
- 没有任何用户密码。
- 加密的保险箱数据库文件通过数据库索引（Database Index）来查找。

秘迹保险箱的数据加密完全依赖加密密钥，没有账户、没有密码，所以加解密不依赖密码认证。

用户对服务端数据库文件的增删查改都必须通过签名校验，签名使用初次使用秘迹时生成的一对 2048 位的 RSA 密钥的私钥，服务器用对应的公钥校验签名。签名用的 RSA 密钥保存在保险箱内。

客户端如何确保数据安全

不论是在客户端还是服务器端，保险箱 KDBX 文件在任何地方保存时都是加密格式存储，只在打开应用时解密到内存。

创建保险箱时：

1. 随机生成一个 AES 密钥；
2. 用 AES 密钥加密一个随机生成的 KDBX 文件高熵值密码，加密后的密码保存到 Android SharedPreferences 或 iOS 本地文件；
3. AES 密钥保存到 Android KeyStore 或 iOS Keychain，以保证密钥只能本应用获取。

打开保险箱时：

1. 应用从 Android KeyStore 或 iOS Keychain 中取出 AES 密钥；
2. 从 Android SharedPreferences 或 iOS 本地文件中取出加密过的 KDBX 密码，用 AES 密钥解密，得到 KDBX 文件密码；
3. 使用此密码打开 KDBX 文件，加载到内存。

锁屏密码

功能说明

- 锁屏密码为应用锁，与加密保险箱数据的密码或密钥无关。
- 为用户提供更灵活的保护方式，如延时锁定应用、切后台立即锁定应用。
- 如果用户设置了密码，在认证身份之前，应用保持在锁定页面，需验证密码后才可以进入应用。

实现方案

- 系统认证方式：根据系统不同，直接调用系统锁屏密码、指纹识别、人脸识别、锁屏密码认证方式，认证通过应用退出锁定页面。

跨屏登录

跨屏登录功能，是在手机上登录网站后，把网站的 cookie 通过 AES-256-CBC 加密，通过 TLS 加密通道发送给电脑浏览器插件，在电脑上用这些 cookie 进行认证。这个过程中，网站密码没有离开过手机，不会在电脑浏览器里面出现，避免了密码被电脑和浏览器上的恶意应用截获的风险，提高了密码的安全性。AES 密钥是每次手机扫码绑定桌面浏览器时在浏览器插件端随机生成的。

悄悄话

悄悄话是一种利用 PKI/CMS 技术实现的端到端加密工具，它使得由一方可以通过任意应用很方便地将信息加密给另外一方，不用担心中间人攻击。我们在此提出一种解决方案，使用户能在其他应用内直接使用该加密工具。该工具通过用接收方的证书加密成 CMS 格式，除非拥有证书对应的私钥，CMS 里面的密文无法解开。这个系统本身需要的基础设施非常少。信息由用户决定通过什么渠道或者应用传播，几乎不受网络条件限制。

通常的通信解决方案，都把认证、加密、消息通道都整合在一起。造成在今天的互联网上，要么不是端到端加密，要么功能不够强大，系统不够稳定，用户在便捷性和安全性上必须做一个选择。

比如互联网上的即时聊天工具，都需要借助中心服务器作为可信赖的第三方来中转，现在仍然有大量通信应用没有使用端到端加密。虽然这类应用在绝大多数情况下都运作良好，但是仍然内生性地存在可被中间人攻击的弱点。我们无法确定消息完全安全的传递，因为服务器总是可能被黑客攻击，从而造成信息泄漏甚至篡改。中心服务器的存在，也增加了用户对特定通信系统的依赖。很多聊天系统联系人关系保存在服务端，如果服务端出于任何原因丢失账号相关信息，用户需要用新账号或者新系统和所有联系人重新建立联系。

所以，我们非常需要这样一种端到端的加密工具，它基于密码学原理而不基于任何其他通信通道，使得任何有联系的双方，能够在任意通信应用中对消息很方便地进行高强度加密，杜绝中间人攻击的可能，从而保护通信双方的通信隐私。只要两端的私钥没有泄漏，该系统就是安全的，无论密文是通过什么通道传递的。

不绑定个人信息

开始使用悄悄话时，用户选择服务端下发的随机生成的 2 ~ 4 个字的昵称，这是悄悄话的唯一 ID。用户无需绑定邮箱、手机号等个人可识别信息。

服务器不参与用户之间的关系建立，所以用户之间的联系需要自行建立。我们服务器只存储密文，不存储悄悄话好友之间的联系关系。

端到端加密

选定昵称后：

1. 本地生成一个 2048 位长度的 RSA 私钥
2. 客户端向秘迹 CA 服务器请求昵称对应的证书
3. 如果这个昵称还没被占用，秘迹服务器即签发并下发悄悄话证书

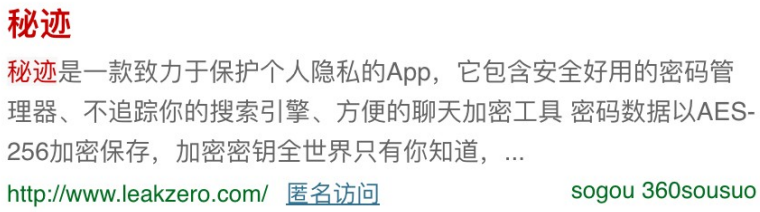
发送方通过接收方公钥把明文信息加密成 CMS 密文消息，密文在服务器端生成短链接。接收方根据短链接获得 CMS 密文，用自己私钥解密。服务器端只保存消息的密文和短链接地址。悄悄话私钥只存在客户端保险箱内，所以服务器无法知道任何一条明文聊天内容。

加密键盘

秘迹加密键盘（暂时只支持 iOS）不发送任何用户输入的文字到手机之外的地方。键盘使用和秘迹应用内悄悄话功能相同的密钥和算法进行文字加解密，明文输入文字始终不会离开客户端。

隐私搜索

用户点击秘迹搜索结果标题链接，浏览器不会请求其它搜索引擎服务，而是通过秘迹服务获得最终的搜索结果网页，跳转并打开结果。



秘迹搜索不在浏览器记录任何 cookie，不在服务端存任何用户信息，不记录用户 IP 地址和搜索关键字。每次搜索后，服务器上记录的日志如下：

```
200 23/Oct/2018:21:36:58 +0800 mijisou.com
```

只有搜索发生的时间和 HTTP 请求状态码，没有记录其它任何信息。

匿名访问

用户点击秘迹搜索结果中的“匿名访问”，浏览器会通过秘迹代理服务直接获取最终搜索结果网页内容，并展示出来。匿名访问过程用户浏览器始终只访问了秘迹服务，没有访问其它搜索引擎、没有访问搜索结果页面，所以更加隐私。

基础架构安全

为了实现安全的端到端加密，我们绝对不在服务器存储密码明文数据，也不存储任何个人身份信息。

传输安全

从秘迹客户端到服务器的所有传输，都使用 TLS 加密。

服务器存储了什么

传统的互联网企业，存储了大量的明文用户数据在服务器。这些明文数据成为了黑客持续的攻击目标。更严重而又被人忽略的，是公司的数据库管理员可能对数据的复制、篡改等滥用行为。秘迹保险箱在安全设计中，始终把所有人的恶意行为都考虑进去了。所以我们在服务器不存储任何明文用户数据和个人身份信息。

我们服务器存储的数据主要由如下几种：保险箱 KDBX 文件（用户密钥加密）、数据文件索引、悄悄话花名、证书、悄悄话短链接和对应的 CMS 密文内容。服务器上不存储悄悄话的发送方和接收方。

如果我们的服务器被黑了，黑客拿到的数据里面，没有任何明文密码，没有任何明文聊天记录，没有任何个人信息。黑客要想得到明文，需要破解 AES-128 强度的恢复密钥。

如何管理 CA

悄悄话用户的证书通过服务器上的悄悄话 CA（Certificate Authority）签发，该 CA 由我们自签发的根证书签发。CA 私钥做了特殊物理加固和监控。悄悄话服务器通过主机权限控制（Host-Based Access Control）限制了有限的运维人员才可以访问该主机，CA 的私钥只有 root 用户可以访问。

秘迹 Android 应用签名的私钥也在一台单独的离线计算机上存储和使用，硬盘全盘加密，全公司只有两个人可以访问该私钥。该私钥也做了特殊物理加固和监控。

根证书在一台全盘加密的离线服务器上生成，该服务器永不上网。根证书的私钥用 AES-256 加密存储，只在签发证书的时候解密。签发证书只通过 OpenSSL 命令，所以私钥的明文永不落地。根证书的私钥加密密码用秘迹保险箱分片保管在三个人手里，需要其中的不少于两个人提供各自的密码，方可解开私钥，进行证书签发操作。

更新记录

V1.0.0 2019-01-25

首次发布。