Atitit 第11章 安全通信：自由的技术 摘要

元英文有40多页word文档

目录

[1.1. 解开安全消息传递的复杂性 1](#_Toc29105)

[1.2. 改造邮件存储 2](#_Toc623)

[1.3. 解密的持久性 4](#_Toc30409)

[1.4. 看不见的手移动 6](#_Toc14828)

[1.5. 速度很重要 8](#_Toc5147)

[1.6. 个人权利的通信隐私 8](#_Toc20027)

[1.7. 入侵文明 9](#_Toc27521)

## 解开安全消息传递的复杂性

虽然为世界带来安全的通信能力是一个非常好的主意

为了保护个人人权（稍后会详细介绍），正确处理是一个技巧

比看起来更艰巨的任务。原则上，公钥密码系统可以促进临时

安全的通信，但实际的实现往往是不必要的复杂

与有关谁将使用此类系统的实地现实脱节，以及

如何。**164**

第十一章

基于公钥的实际实现中要解决的根本问题

密码学是密钥认证。要向某人发送加密消息，您需要

她的公钥。如果您可能被诱骗使用错误的公钥，您的隐私

消失。

密钥认证问题有两种截然不同的方法。

传统的公钥基础设施 (PKI) 方法，通常基于 ISO 标准

dard X.509，取决于受信任的第三方证书颁发机构 (CA) 系统，

并且在很多方面根本不适合满足临时网络中用户的实际需求

作品。\* PKI 实现在更加结构化方面取得了显着的成功

域，例如企业 VPN 和安全网站的身份验证，但具有

在现实世界的异构电子邮件环境中进展甚微。

另一种方法以最流行的基于公钥的消息传递安全为例

今天使用的 rity 解决方案：Phil Zimmermann 的 PGP 及其后代，现在正式化为

IETF OpenPGP 协议。OpenPGP 保留了灵活性并从根本上说

通过促进分布式密钥认证来追踪公钥密码学的性质

通过“信任网络”，而不是依赖于集中的、等级制度

CA，正如 PKI 方法所做的那样（包括 OpenPGP 的主要竞争对手，S/MIME）。不是 sur

令人惊讶的是，在流行的电子邮件客户端中几乎无处不在的 S/MIME

尽管电子邮件客户普遍缺乏理解，但用户群比 OpenPGP 小得多

对 OpenPGP 的大力支持。

但是信任网络方法依赖于用户建立自己的信任链

认证和验证公钥有其自身的问题。其中最主要的是

确保用户了解如何使用信任网络来实现相互关联的挑战

验证密钥，以及需要达到临界数量的用户以确保

任何两个用户都可以轻松找到彼此之间的信任路径。

如图 11-1 所示，在信任网络实施中，没有第三方是任意的

指定为“受信任”。每个用户都是自己最信任的认证机构，

并且可能会为了验证密钥而向其他人分配不同级别的信任。你

如果密钥由您或完全信任的另一个人直接认证，则认为该密钥有效

由您来认证密钥，或由用户定义的人数，每个人都是部分

受您信任来认证密钥。

因为信任网络方法不会尝试将密钥认证外包

以 PKI 方法的方式，用户必须在建立他们的信任网络和

确定公钥的真实性。这将可用性考虑放在首位

基于 OpenPGP 的安全消息系统设计的中心。

\* 传统 PKI 的缺点已由 Roger Clarke 简明总结，*网址*为*http://*

## 改造邮件存储

将用户邮件存储在纯*mbox*文件中适用于第一个原型，但生产系统

tem 需要能够更有效地访问和更新单个消息

允许单个平面文件邮箱。我也想朝着非常重要的目标前进

为容错提供邮件存储复制。

\* 在 Perl 中，引用通过bless关联到类时成为对象，因此“受祝福的引用”是

只是一个对象的 Perlish 术语。安全通信：自由技术

**175**

可用性考虑也对邮件存储提出了一些要求。在加密

nite，与大多数电子邮件客户端不同，有关消息的 MIME 结构的信息将是

在消息列表中对用户可见。这将使用户可以在视觉上

直接在消息列表中识别哪些消息被加密和/或签名。可用性

消息列表中有关消息部分的信息的能力也将使用户能够

直接打开消息子部分。消息部分显示为最右侧的图标

消息列表视图栏，如图11-4所示。

为了实现这种视觉反馈，邮件商店需要有效地提供准确的

有关消息列表的 MIME 结构的信息。进一步的并发症是

OpenPGP/MIME 规范允许 MIME 部分嵌套在已签名中的事实

和/或加密部分，因此只有 OpenPGP/MIME-aware 邮件存储可以返回 accu

对有关加密或签名消息的 MIME 结构的信息进行评级。

所以我决定基于 Mail::Folder 模块实现一个基于 SQL 的邮件存储

后端具有 IMAP4rev1 服务器的大部分功能。这个系统的核心是

Mail::Folder::SQL 类，基于 Mail::Folder 并使用 Persistence::Object::Postgres。这

当 IMAP 还没有获得太多关注时，它又回来了。我选择不使用现有的

IMAP 服务器作为邮件存储，因为我预计需要大多数 IMAP 的一些功能

服务器没有很好的支持，例如邮件存储复制和检索能力

无需检索即可获得有关 MIME 消息结构的详细信息

并解析整个消息。

即使某些 IMAP 服务器可能满足我的需求，我也不想要 Crypto

nite 依赖并绑定到任何特定 IMAP 服务器的功能

执行。总而言之，结果证明这是一个很好的决定，尽管它确实导致了

在代码上花费了大量精力，这些代码后来被降级为不太重要的角色

系统。

邮件存储复制是使用我编写的两个 Perl 模块进行的：Replication::Recall

和 DBD::Recall，它使用了 Eric Newton 的 Recall 复制框架（*http://www.*

*容错.org/recall*）跨多个服务器复制数据库。这个想法是使用

以此作为原型，并在未来定制构建一个新的数据库复制系统。

随着加密、数据库和邮件存储后端的改进，以及一个新的、

更干净的主题，Cryptonite 的第一个内部测试版于 2001 年 10 月上线。

由许多不同技能水平的用户进行测试，其中一些人甚至将其用作他们的主要

*图 11-4*。*包含第***176***部分的消息列表*

第十一章

邮件客户端。内部测试版期间的可用性测试表明，新手用户能够

成功生成和导入密钥，以及发送和读取加密和签名的我

贤者无大碍。

## 解密的持久性

加密邮件客户端的一个基本功能是能够保留解密的消息

在用户会话期间以解密形式提供。一个安全的邮件客户端

缺少此功能会变得非常烦人且使用效率低下，因为它需要输入

每次您想阅读加密的密码时都需要长密码并等待解密

消息或在加密消息中搜索。

Cryptonite 中先前解密的消息的持久性是通过创建

一个新的 Mail::Folder 类，基于 Mail::Folder::SQL。Mail::Folder::Shadow 会删除

如果邮件在影子中有副本，则网关邮箱访问*影子文件夹*

文件夹; 否则，它将访问底层（或*隐藏*）文件夹。

通过这种方式，解密的消息可以在会话期间保存在影子文件夹中。

活着，并且几乎不需要修改代码来添加持久解密，其他

而不是在任何使用 Mail::Folder::SQL 的地方插入 Mail::Folder::Shadow 模块。

Mail::Folder::Shadow 使用一个简单的、可调整的委托表来实现它的魔力：

我的 % 方法 =

qw (get\_message 1 get\_mime\_message 1 get\_message\_file 1 get\_header 1

get\_mime\_message 1 mime\_type 1 get\_mime\_header 1 get\_fields 1

get\_header\_fields 1 refile 1 add\_label 2 delete\_label 2

label\_exists 2 list\_labels 2 message\_exists 1 delete\_message 5

同步 2 删除 2 打开 2 set\_header\_fields 2 关闭 2 DESTROY 2

get\_mime\_skeleton 1 get\_body\_part 1);

Mail::Folder::Shadow 将方法调用委托给影子文件夹，原件

inal 阴影文件夹，或两者兼而有之。Perl 强大的AUTOLOAD功能，它提供了一个机制

anism 来处理类中未明确定义的方法，是一种简单的方法

完成此委托，同时还提供了一种在运行时进行调整的简单机制

如何处理不同的方法。

必须检查影子存储的方法，例如get\_message和get\_header ，是 del

如果相关消息存在于 shadow 文件夹中，则归入 shadow；除此以外，

它们被委派给原始的隐藏文件夹。其他方法，例如add\_label和

删除（删除文件夹），需要分派到阴影和阴影

欠文件夹，因为这些消息必须更改原始文件夹的状态，以及

的影子文件夹。

还有其他方法，例如delete\_message ，可以通过数组 ref 接受消息列表

伦斯。消息列表中的某些消息可能会被隐藏，而其他消息可能不会。

Mail::Folder::Shadow 的AUTOLOAD通过从 mes 构建两个列表来处理这些方法

传递给它的 sage 列表，一个隐藏消息和一个非隐藏消息。它

然后在 shadowed 和 shadow 文件夹上调用该方法以获取

阴影，并且仅在阴影文件夹中用于未显示的邮件。安全通信：自由技术

**177**

所有这一切的实际结果是*cmaild*可以继续像以前一样使用文件夹

之前，并在会话期间将解密的消息存储在影子文件夹中。

Mail::Folder::Shadow 中有一些额外的方法可以启用它，包括

update\_shadow ，用于将解密后的消息保存在 shadow 文件夹中；

delete\_shadow ，用于根据用户请求删除单个阴影消息；和无影，

用于在会话终止前删除影子文件夹中的所有邮件。

Mail::Folder::Shadow 可以提供解密消息的持久性

会话并在加密消息中实现搜索——这两个基本功能来自

从用户的角度来看，但很少在当前一代的 OpenPGP 兼容中实现

电子邮件系统。

## 看不见的手移动

到 2004 年年中，Neomailbox 已经成立一年，并吸引了不少付费客户。

Cryptonite 开发被搁置了一段时间，而我正在开发各种

Neomailbox 服务的各个方面以及其他一些我迫不及待的项目

开始吧。

但是进入市场很棒，因为它带来了市场力量，从竞争到

用户反馈，影响开发过程，并帮助锐化和澄清先验

关系。客户的请求和查询帮助我与客户保持密切联系

用户和市场想要。满足市场需求是应用程序代码的方式

毕竟是商业意义上的变美，所以与市场的互动就变成了

开发过程中不可或缺的重要组成部分。

Cryptonite 被设计为易于维护和修改，正是因为我知道

在某个时候，它必须开始以新的方式发展，既是为了应对，也是为了

预期客户想要什么。在市场上让我看到了

新兴需求：很明显，IMAP 是远程邮箱访问的未来。

IMAP 有很多吸引人的功能，使它成为一个非常强大和实用的邮件访问

协议。其中最重要的一项是能够访问同一个邮箱

使用多个客户端，这变得越来越重要

计算设备。典型的用户现在拥有台式机、笔记本电脑、PDA 和手机，

都可以访问她的邮箱。

这带来了一个小问题，因为我已经为 Cryptonite 实现了一个完整的邮件存储，并且

它不是基于 IMAP 的。有两种前进的方式：要么实现一个完整的 IMAP 服务器

基于 Cryptonite 邮件存储（一项大工作），或修改 Cryptonite 以使其能够使用

IMAP 邮件存储作为后端。事实上，第二个必须以任何一种方式完成。

再次，选择降低系统的复杂性，并专注于其主要目的，我

决定不将 Cryptonite 邮件存储开发为成熟的 IMAP 服务器。相反，我

将其修改为缓存机制，缓存 MIME 骨架（只是结构

信息，不包括用户列出的多部分 MIME 消息的内容，以及安全通信：自由技术

**183**

用户阅读的整个消息，以便用户下次打开她阅读的消息时

之前，Cryptonite 不需要返回 IMAP 服务器再次获取它。

这给了我两全其美的好处。Cryptonite 可以反映 IMAP 的内容

邮箱，同时提供每封邮件的确切 MIME 的完整信息

结构，以及能够在影子文件夹中保留解密的消息

支持 Cryptonite 邮件存储。

对代码的修改很简单。每当用户点击阅读

不在缓存中的消息，Cryptonite 将其缓存在相应的 Mail::Folder::

影子文件夹：

以类似的方式，为用户列出的所有消息缓存 MIME 骨架

通过消息列表视图。其余的代码继续像以前一样工作，通过操作

在缓存上进行所有读取操作。现在我们有了 IMAP 兼容性，无需 com

承诺我的邮件商店提供的功能或大量修改主要代码。

邮件存储复制需要再次工作，因为从 Mail:: 切换

Folder::SQL 到邮件存储的 IMAP 服务器意味着 Replication::Recall 不能

用于复制。但无论如何，Replication::Recall 并不是最优雅或最容易的

实现复制系统，Recall 库已经用 Python 重写，mak

无论如何，将我的 Perl 接口连接到早期的 C++ 实现已经过时了。**184**

第十一章

事后看来，我花了很多时间在复制功能上，这必须是

报废了，我可能最好不要在那个时候打扰复制

阶段。另一方面，它确实教会了我很多，当我开始学习时会派上用场

再次实施复制。

市场力量和不断变化的标准意味着应用软件总是在不断发展，

从程序员的角度来看，这些代码的大部分美妙之处肯定在于

使代码适应不断变化的需求是多么容易。Cryptonite 的对象

面向架构使得轻松实施重大修订成为可能。

## 速度很重要

使用 Cryptonite 邮件存储，性能非常快，大多数邮件存储

操作与邮箱大小无关。但是随着切换到 IMAP，我注意到

大型邮箱的一些主要减速。一点剖析表明，perfor

钉头锤的瓶颈是纯 Perl Mail::IMAPClient 模块，我曾经用它来实现

IMAP 功能。

一个快速的基准脚本（使用 Benchmark 模块编写）帮助我检查

是否是另一个 CPAN 模块 Mail::Cclient，它与 UW C-Client 接口

库，比 Mail::IMAPClient 更有效率。结果清楚地表明，我会

使用 Mail::Cclient 重做 IMAP 代码：

在编写之前，我可能应该考虑对不同的模块进行基准测试

使用 Mail::IMAPClient 的代码。我最初避免使用 C-Client 库，因为我想要

使构建过程尽可能简单，Mail::IMAPClient 的构建过程是

绝对比 Mail::Cclient 简单。

幸运的是，从前者到后者的转换通常非常简单。

对于某些操作，我注意到 IMAPClient 可以比 C-Client 做得更好

没有太多的性能损失，所以 Cryptonite::Mail::Service 现在使用两者

模块，每个模块都做它擅长的事情。

像Cryptonite这样的程序当然永远不会“完成”，但是代码现在已经成熟了，

健壮、功能齐全且高效，足以达到其目的：提供数千种

并发用户提供安全、直观和响应迅速的电子邮件体验，同时帮助他们

有效保护他们的通信隐私。安全通信：自由技术

## 个人权利的通信隐私

我在本章开头提到了使安全通信技术

广泛使用的系统是保护个人权利的一种非常有效的手段。作为这个认识

nition 是 Cryptonite 项目背后的基本动机，我想以几个结束

关于这一点的观察。

除其他外，加密技术可以：\*

•

为活动家、非政府组织和记者提供强有力的救生保护

压制性国家。†

•

保持被审查的新闻和有争议的想法的可传播性。

•

保护举报人、证人和家庭暴力受害者的匿名性。

•

对于甜点，通过实现自由无拘无束的交流来催化测地线社会

全球范围内的创意、商品和服务。

被称为 Cypherpunks 的五花八门的黑客团队一直在开发隐私

多年来增强软件，旨在增强个人自由和个性

数字时代的主权。一些加密软件已经是

今天的世界是如何运作的。这包括 Secure SHell (SSH) 远程终端软件

件，这对于保护 Internet 基础设施至关重要，以及安全套接字

层 (SSL) 加密套件，可保护在线商务。

但是这些系统针对非常具体的需求：安全的服务器访问和安全的在线信用

卡交易，分别。两者都关注确保人机交互

。专门针对人与人之间的更多加密技术

需要在未来几年采取行动以对抗不断发展的威胁

无处不在的监视（“导致文明的快速终结” ‡ ）。

一个易于使用、安全的网络邮件系统是一种使能技术——它使

历史上第一次，印度之间安全和私密的长途通信

全球各地的视频，从不需要亲自见面。

## 入侵文明

这台计算机具有无限而微妙的复杂性，以至于它包含有机生命本身

操作矩阵——地球及其承载的文明——可以重新编程

用简单的代码片段开发强大的方法，将人类文化还原并重新布线

社会本身的操作系统。

\* 见*http://www.idiom.com/~arkuat/consent/Anarchy.html*和*http://www.chaum.com/articles/Security\_*

*Wthout\_Identification.htm。*

† 参见*http://philzimmermann.com/EN/letters/index.html。*

‡ Vernor Vinge，*天空深处。*Tor 书籍，1999. **186**

第十一章

代码已经多次改变了世界。考虑通过以下方式实现的医学进步

基因测序软件，商业软件对大型企业的影响和

小型企业一样，工业自动化软件带来的革命和

计算机建模，或互联网的多次革命：电子邮件、网络、博客、

社交网络服务、VoIP……显然，我们这个时代的大部分历史都是关于

软件创新。

当然，代码和任何技术一样，可以双向切割，要么增加要么减少

社会中的“暴力回归” \* ，增加了侵犯隐私技术的效力

一方面，给予暴君更有效的审查工具，或增强

另一方面促进个人权利。任何一种代码都破解了最核心的

人类社会本身，通过改变基本的社会现实，例如自由的可持续性

演讲。

有趣的是，即使使用特定的技术，如公钥密码学，imple

选择的心理可以显着改变文化现实。例如，一个基于 PKI 的

实施重新强加了威权属性，例如集中的等级制度和

对一项技术的识别要求，其全部价值可以说在于缺乏

那些属性。尽管如此，PKI 方法提供的密钥认证比

进行信任网络实施（这也不会淡化其他重要功能

公钥密码学，例如分布式部署）。

我认为作为代码的编织者，这在很大程度上是专业人士的道德责任

语法师不仅要追求我们的代码在设计和实现上的美观，

而且它的结果是美丽的，在更大的社会背景下。这就是为什么我发现免费

dom 代码太漂亮了。它将计算技术用于最崇高的用途：保护

人权和人类生活。

法律和国际人权条约只能在保护个人

权利。历史表明，这些很容易绕过或忽略。在另一

一方面，密码系统的数学在仔细实施时可以提供实践

个人权利和公开表达的坚不可摧的盾牌，最终可以设置

世界各地的人们可以自由地交流和交易隐私和自由。

实现这个全球性的、受数学保护的开放社会在很大程度上取决于我们，众神

机器的。

---end