论文笔记

题目：Comparing the Usability of Cryptographic APIs

出处：IEEE Symposium on Security and Privacy 2017

作者：Yasemin Acar, Michael Backes, Sascha Fahl, Simson Garfinkel, Doowon Kim, Michelle L. Mazurek and Christian Stransky

单位：CISPA, Saarland University; National Institute of Standards and Technology; University of Maryland, College Park

原文：https://saschafahl.de/papers/cryptoapis2017.pdf

相关材料： Video

一、背景

在许多应用程序中都存在着潜在的密码学算法误用问题。传统观点认为，许多密码学算法误用问题是由加密的应用程序编程接口（API）引起的，这些接口过于复杂，缺乏安全性，或者文档不完善等。许多研究人员使用静态和动态分析技术来识别和调查源代码或二进制文件中的密码学误用问题。这种方法对于说明密码学误用问题的普遍性以及识别实践中最常见的错误种类非常有价值，但它不能揭示根本原因。安全社区的传统观点认为，这些错误在很大程度上会增加，因为对于非密码学专家来说，很难正确的使用这些密码库。特别的，密码库和应用程序编程接口（API）被广泛认为是复杂的，有许多令人困惑的选项和很差的默认选项（有时候甚至是不安全的默认选项）。

二、提出的方法以及解决的问题

为了解决上述问题并了解如何构建有效的未来密码库。作者进行了一个受控实验，从GitHub招募256位Python开发人员，让他们使用五种不同密码库（cryptography.io、 Keyczar、 PyNaCl、 M2Crypto 和 PyCrypto）之一，并尝试使用对称和非对称加密的常见任务。检查他们的结果代码的功能正确性和安全性，并将他们的结果与他们自己报告的关于他们分配的密码库的使用情况进行比较。以调查密码学算法误用问题的根本原因，并比较不同的密码库的API接口的可用性。此外，通过了解这些实验者开发人员成功使用密码库的情况和错误使用密码库的情况的根本原因，为未来的密码库设计制定蓝图。

三、技术方法

作者设计了一个在线研究课题，以比较开发人员如何使用不同的加密库快速编写正确、安全的代码。作者在GitHub上招募了具有Python开发经验的的256个开发人员。参与者被分配到的任务是：使用对称或非对称密钥加密完成一小组编程任务，并且使用五个Python加密库中的一个。在10个候选的任务中随机的分配一个给每一个试验者，并为确保分配均衡，每分配一个任务给实验者的时候都记录下该任务。然而，在实验的过程中，由于各个任务的参与者的退出率（任务没有做完就退出了）差异很大，所以作者对随机分配进行加权以平衡每一种任务的退出率。在每种条件下，任务顺序是随机的。参与对称加密的试验者要么被给予密钥生成，要么被给予加密/解密任务。而参与非对称加密的试验者被分配了密钥生成任务、加密/解密任务和证书验证任务。

四、实验评估

经过试验发现，简化密码库的API接口，事实上并不能保证密码库的可用性更强。相反，从试验者所写的代码中可以看出，密码库文档的质量，特别是密码库的示例代码是否可以在因特网上获得，以及是否在所提供的文档之内，这些条件更加能够表明一个密码库的可用性。另一方面，简化的API接口看起来似乎促进了试验者能写出更好的安全代码，减少开发人员必须做出的选择（例如，密钥大小或加密操作模式）的数量也会减少他们选择不正确参数的可能性。此外，一些有安全背景的开发人员，写出安全的代码的概率更加大，这也是属于意料之中，因为有安全背景的开发人员，至少是知道密钥的长度以及加密模式的选择等。总之，实验结果不是很理想，有20%的参与者认为他们的代码是安全的，但是实际上是不安全的。结果表明，想要提高新密码库的安全性，密码库就应该提供简单方便的接口，并且要是确保该密码库支持广泛的常见任务，并提供可访问的文档，并且在文档中包含安全、易用的代码示例。

密码算法的安全性评估：对于对称加密算法，作者将ARC2、ARC4、Blowfish、（3）DES和XOR加密算法评为不安全密码算法，而将AES评为安全的密码算法。此外，作者把ECB加密模式评为不安全的加密模式，而把Cipher Block Chaining (CBC)、Counter Mode (CTR) 和 Cipher Feedback (CFB)加密模式评为安全的加密模式。对于非对称加密算法，作者只把使用 OAEP/PKCS1填充方式评为安全的填充方式。

实验结果还表明：在密钥生成、加密、解密和认证这几个任务当中，加密被认为是最简单的。对于使用对称加密算法的试验者，85.2%的试验者完成加密任务，而在这85.2%的试验者当中，只有70.1%的试验者所实现的代码是安全的；而对于分配到非对称加密算法的试验者，72.0%的试验者完成加密任务，而在这72.0%的试验者当中，只有78.8%的试验者所实现的代码是安全的。而在整个实验过程中，证书认证被认为是最难的，只有22.4%的非对称加密试验者能够完成证书认证的任务，很不幸的是，在这22.4%的试验者当中，没有一个试验者的代码是安全的。

五、优缺点

优点：

- 试验者数量相对较大，使得试验结果更加具有说服力，更加可靠。
- 这个实验是一个在线的实验，不需要额外的基础设施，因此，实验的代价比较小。
- 在线实验的目标对象地域分布广泛，有来自全世界各地的试验人群。

缺点：

- 由于试验者是自愿参与本实验的，所以可能会导致试验结果不太完好，有些试验者做到一半就退出了，而有些又不太认真对待这种试验，所以这些因素可能在一定程度上影响实验的结果。
- 由于只对Python开发人员进行试验，而没有对其它语言的开发人员做相应的实验，就可能使得结果没有这么完美。
- 由于密码库的选择是按照这些密码库的流行性与可用性进行选择的，因此不具有随机性，可能使得试验结果有一定的偏差。

六、个人观点

作者使用在线调查报告的形式做这个实验，而目标对象是Github上的、相对活跃的Python开发人员，实验的基础设施要求不高，但是实验的工作量比较大。作者首先需要找到Github上的Python开发人员，然后经过筛选，给目标试验者发送邀请邮件，邀请他们参加这个试验，但是由于这个实验是自愿参与原则，因此，很多开发人员不愿意或者没有时间参与这个实验。这个是进行该实验的第一个难题，也是最主要的难题。其次，对于密码库的设计，存在很大的问题是：很多时候，这些密码库的文档写的很差，甚至很难找到相应的示例代码，最糟糕的是，即使是官方文档中给出的示例代码，也有存在不安全的用法，这就给使用这造成很大的灾难。因为对于普通的非专业用户，或者是没有密码学相关经验或者背景的开发者，他们一般会直接按照文档中的示例代码或者网络上找到的示例代码的使用方式来使用，这就很可能会给他们开发出来的应用程序带来潜在的漏

洞。因此，设计一个良好的密码库，不仅需要有简单易懂的API接口，还需要有一份通俗易懂的说明文档提供给使用者，此外，文档中应当包含各种应用场景下的示例代码，保证示例代码不存在密码算法误用，有默认参数或默认选项的，要保证默认参数或默认选项是安全的，这样设计出来的密码库才能更易于使用。