基于硬件加密卡技术的 RSA 加密引擎的局部封装

徐 敏
(国网电力科学研究院, 江苏 南京 210003)

摘 要：目前主流的 VPN 是通过 SSL 协商进行认证的，其在调用 Openssl 的加密引擎过程中带来一个问题，即进行数据加密所使用的公钥不是自身的密钥，于是加密引擎不仅要有对私钥解密进行硬件实现，还要有对公钥加密实现原有算法调用的技术。本文介绍了目前主流的 Openssl 实现硬件 RSA 加密引擎仅对私钥解密数据进行调用，而对于公钥加密数据不调用的局部封装方法，有着重要的实用价值。

关键词：VPN；SSL 协商；Openssl 引擎；RSA 加密
中图分类号：TP393 文献标识码：A 文章编号：1674-7720(2011)20-0003-03

The local package of RSA encryption engine based on hardware-based encryption card technology

Xu Min
(State Grid Electric Power Research Institute, Nanjing 210003, China)

Abstract: VPN is currently the mainstream for certification through the SSL negotiation, the encryption engine in the calling process Openssl creates a problem. Namely, the public key used to encrypt the data itself is not the key, so not only has a private key encryption engine decrypt the hardware implementation, but also for public key encryption algorithm to achieve the original call technology. This article describes the current mainstream hardware Openssl RSA encryption engine to decrypt the local data encapsulation methods in which private key data calls only, and does not call for public-key encryption. It has important practical value.

Key word: VPN; SSL negotiation; Openssl engine; RSA encryption

目前，国家电网公司信息内外网边界的各类业务接入对象已经越来越多地采用多种接入方式。传统的内网专线模式已经远远不能满足日益增长的办公需要。随着智能电网的普及，电网内部专线资源也逐渐对外网的接入提出了迫切的需求。因此对满足内外网数据安全交互的 VPN 技术提出了更高的要求。Openssl 以其开源、高效、安全的加密方式在数据加密的过程中发挥了巨大作用。但是由于 SSL 协议的特性，公钥将会在服务器和客户端直接传递，即进行数据加密所使用的公钥是由对方传递来的，而不是自身的密钥。这样也带来一个问题，传统 openssl 加密引擎必须实现对于 RSA 引擎的局部调用，即只对需要 RSA 私钥解密的数据进行硬件引擎的调用，而对于 RSA 公钥加密的数据采用原来的软加密方式不经过硬件引擎调用。

1 硬件 RSA 加密引擎主要技术

1.1 Openssl 硬件加密引擎封装技术[1]

Openssl 硬件引擎(Engine)能够使用户比较容易地将自己的硬件加入到 Openssl 中去，替换其提供的软件算法。一个 Engine 提供了密码计算中各种计算方法的集合，它用于控制 Openssl 的各种密码计算。引擎机制的目的是为了使 Openssl 能够透明地使用第三方提供的软件加密卡或者硬件设备进行加密。换句话说就是对 Openssl 进行了一种“欺骗”，对于指定的一种算法，Openssl 会自动用引擎内的方法对其实现加解密，然后默认其是由该算法进行加密的，而对于引擎内的具体实现不去关心。因此可以对引擎进行自定义式的实现，比如用硬件加密卡接口对其进行数据的加解密等，从而实现硬件加密卡与软件数据流之间的交互。

1.2 RSA 加密技术[2]

RSA 算法是一种非对称密码算法。所谓非对称，就是指该算法需要一对密钥，使用其中一个加密，而需要用另一个才能解密。RSA 的算法涉及三个参数：n、$e1$、$e2$。其中，n 是两个大质数 p、q 的积，n 的二进制表示钥所占用的位数，就是所谓的密钥长度。$e1$ 和 $e2$ 是一对相

关的值，$e1$ 可以任意取，但要求 $e1$ 与 $(p-1)\times(q-1)$ 互质；再选择 $e2$，要求 $(e2\times e1)\bmod((p-1)*(q-1))=1$。($n$ 及 $e1$)，(n 及 $e2$)就是密钥对。RSA 加解密的算法完全相同，设 A 为明文，B 为密文，则：$A=B^{e1} \bmod n$；$B=A^{e2} \bmod n$；$e1$ 和 $e2$ 可以互换使用，即：$A=B^{e2} \bmod n$；$B=A^{e1} \bmod n$。

2 RSA 硬件加密引擎局部封装技术

2.1 RSA 加密引擎[3]

RSA 加密引擎是在 vpn 进行身份认证 SSL 协议握手时对私钥解密部分进行硬件加密卡的调用，从而提高设备数据处理的效率。由于证书只能存放于加密卡内无法取出，可以做到一人一卡从而大大增加设备的安全性和防黑客攻击的能力。如图 1 所示，vpn 采用 Openssl 框架进行加解密运算，用户通过硬件加密 key 专用接口或通用 CSP 接口，使用硬件加密 key 对数据进行加解密，而不是采用 Openssl 默认的软加密方式进行加解密。从而可以对程序的安全性有更好的保障，黑客在没有获得加密 key 的情况下，无法对数据进行加解密运算，也就保证了程序在应对黑客攻击的过程中更加具有安全性。

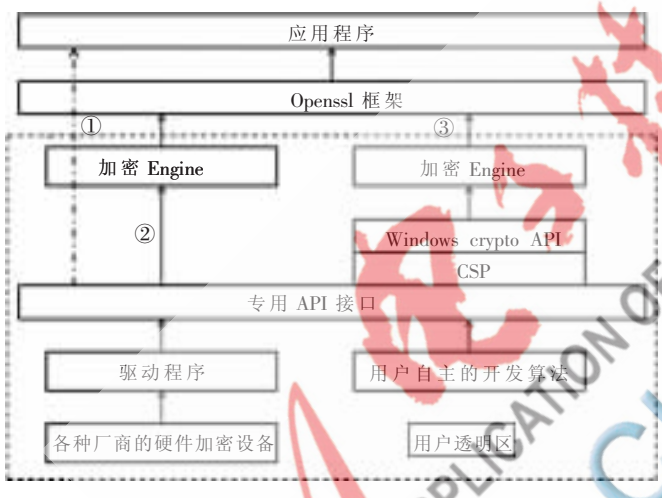

图 1 硬件加密引擎技术

2.2 传统加密引擎封装方法[4]

传统加密引擎封装主要通过指定一个加密算法，对其进行替换处理，用来“欺骗”Openssl，例如图 2 所示采用 AES128 算法进行替换，每当 Openssl 需要进行加解密并且判断得知所采用的算法是 AES128 算法后，会自动放弃使用软件加解密的方式。而通过接口调用硬件中的加解密函数，由于 Openssl 本身并不知道硬件中所采用的算法究竟是什么算法，因此相当于对 Openssl 进行了某种意义上的欺骗。Openssl 默认该硬件加密卡采用了 AES128 算法进行了加解密，但是实际上采用的也许是用户自定义的算法，从而实现了对 AES128 算法的替换。具体流程如图 2 所示。

2.3 只封装私钥的 RSA 加密引擎

在实际运用过程中，由于公私钥不是同一台电脑产生的，在 vpn 中客户端使用的公钥加密的公钥

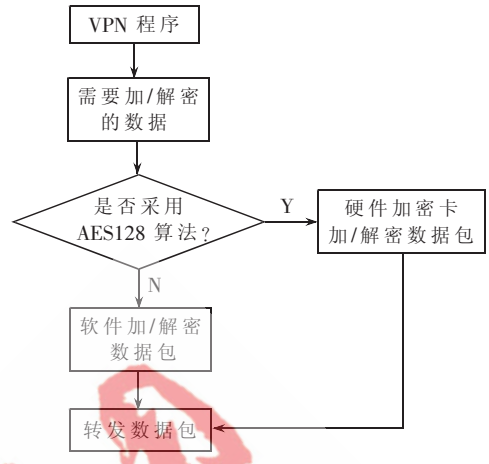

图 2 传统加密引擎封装

是来自于服务端，因此在硬件 key 中不可能预先保存公钥信息，对于公钥加/解密不能一并用硬件引擎实现。由于 Openssl 是通过识别 AES128 算法来进行对硬件引擎的调用，其本身并不能识别出此次加/解密所用的密钥是公钥或是私钥，因此公私钥判断并调用不同运算的过程只能在引擎封装中实现。在引擎加密初始化时需要声明如下：

```
name_rsa.rsa_pub_enc = ossl_rsa_meth->rsa_pub_enc;
name _rsa.rsa_pub_dec = ossl_rsa_meth->rsa_pub_dec;
```

其中 ossl_rsa_meth->rsa_pub_enc 表示 openssl 自身的公钥加密算法，即不通过调用硬件引擎的软加密算法，name_rsa.rsa_pub_enc 表示该硬件引擎所要采用的公钥加密算法，第一行代码连起来的作用解释为将 Openssl 自身的公钥加密算法赋值给该硬件引擎所要采用的公钥加密算法。通过此次赋值，硬件再次调用公钥加密算法时，实际上又再次调用了软加密方法，即相当于对硬件加密引擎也进行了一次“欺骗”，重新回到主程序调用硬件加密引擎前的状态。第二行代码与第一行代码作用相似，只不过是将公钥加密变为公钥解密的算法。具体流程如图 3 所示。

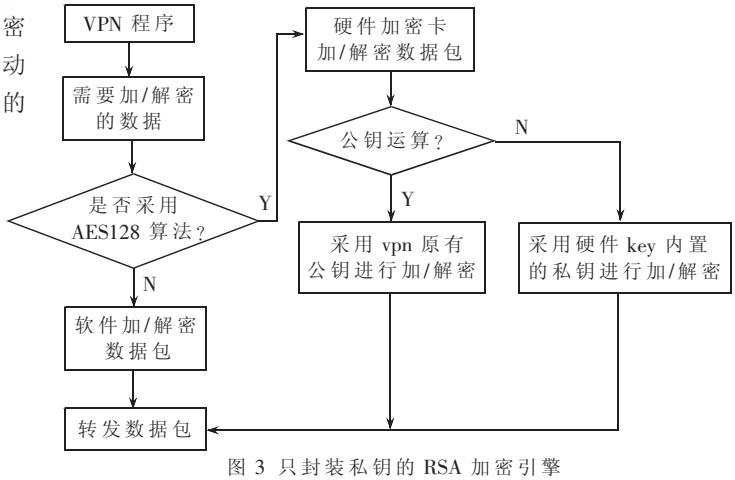

图 3 只封装私钥的 RSA 加密引擎

3 实验结果

本次实验服务器性能如下：linux 64 位操作系统、1 000 MB 以太网卡、4 GB 内存、E5300@2.60 GHz 双核 CPU、高速 PCI 加密卡。

本次实验客户机性能如下：Windows xp32 位操作系统、1 000 MB 以太网卡、4 GB 内存、M520@2.4 GHz 双核 CPU、高速加密 USB key。

实验数据如表 1 所示，采用硬件 RSA 加密方式可以既安全又快速地完成 SSL 握手阶段。如表 2 所示，在大规模海量客户端接入时采用硬件 RSA 加密方式的优势十分明显。

表 1 单个客户端接入实验结果

RSA 加密方式	安全性	公钥运算速度/(ms/次)	私钥运算速度/(ms/次)	认证时间/ms
软加密	低	5	50	200
局部硬件引擎封装	高	3	32	40

表 2 大量客户端并发接入测试结果

同时接入的客户端数量	RSA 加密方式	完成所以客户端认证需要的时间/s
100	软加密	20
	局部硬件引擎封装	14
1 000	软加密	500
	局部硬件引擎封装	342

通过硬件实现 vpn 会话过程中的加解密具有快速、安全、抗攻击等优点，但是传统的硬件封装技术只能对整体算法进行替换封装，对于 vpn 中公私钥不是同一套密钥的情况，传统的封装方法显然不能满足要求，因此对硬件加密引擎进行局部封装。本文提出的方法只封装私钥加解密部分，而对于公钥加解密部分则重新调用原有软加密算法进行加解密，实现了加密引擎封装的灵活可靠性，并很好地解决了 vpn 中公私钥分离的难点，对用硬件加密卡实现数据加密的 VPN 技术有着重要的实用价值。

参考文献

[1] 陈宏标. 超长距离光传输系统设计[D]. 武汉：华中科技大学，2006.

[2] 朱柯嘉，杨青松，徐科，等. 一种 RSA 算法的新型 ASIC 实现[J]. 复旦大学学报：自然科学版，2004，43(1)：16-20.

[3] RIVEST R L.A method for obtaining digital signatures and public-key cryptosystems[D]. Communications of the ACM，1978.12.

[4] 宋磊，罗其亮，罗毅，等. 电力系统实时数据通信加密方案[J]. 电力系统自动化，2004(14)：76-81.

(收稿日期：2011-06-20)

作者简介：

徐敏，男，1984 年生，硕士研究生，助理工程师，主要研究方向：电力信息传输过程的安全加密网关设备研发。