一种基于口令的防窃取私钥保护协议①

王耀民 王立斌 (华南师范大学 计算机学院 广东 广州 510631)

摘 要: 现代密码体制中，加密算法是公开的。因此数据安全取决于用户私钥的保护。目前彩的最为广泛的私钥保存方案是使用用户密码将私钥加密后保存在用户设备上，但是这种记法不能够抵抗攻击者在获取用户设备后，对其使用离线字典攻击。根据对一种现有的能够防窃取的用户设备与服务器联合签名的(S-RSA)协议的分析，提出一种新的基于口令的私钥保护(SS-RSA)协议，该协议通过用户设备与服务器联合对文件进行解密。能够有效的保护用户私钥的安全，并解决了 S－RSA 协议不能够抵抗拒绝服务攻击和用户票据取消后，用户私钥无法恢复的缺点。

关键词: 私钥保护；口令；私钥安全

A Protocol Based on Password Resilient to Devices Capture

WANG Yao-Min , WANG Li-Bin

(Computer School, South China Normal Univercity, Guangzhou 510631, China)

Abstract: The cryptographic algorithm is public in modern cryptography. The security of user's file relies on the protection of the user's private key. Common practice of protecting private key is to encrypt it with a password and store it in the user's device. However, the private key is vulnerable to offline dictionary attack when the device is captured by an adversary. In this paper, we analyze the S-RSA protocol and propose an SS-RSA protocol which can resolve the problem of S-RSA that can't resist the DOS attack and can't get back the user's private key after the user cancels the ticket.

Keywords: private key protection; password; private key security

近几年人们在工作生活和学习中大量的使用便携设备进行在线的数据读写操作，其中涉及了大量的敏感数据，在给人们带来方便的同时，也造成了新的安全问题。由于便携设备的体积和重量都比较的小，攻击者可能会通过某种不正当的手段来获得用户的设备，从而读取用户设备上的数据。传统的作法是将用户私钥用口令加密后存储在用户设备上，但是由于用户选择的口令的信息熵一般比较低，攻击者在获取用户设备后，很容易通过离线字典攻击得到用户的口令，从而取的 S－RSA 密钥保护协议，采取了用户与服务器联合签名的协议。这种协议将用户的私分割成两部分，分别由服务器和用户设备保存。能够有效的保证用户获取用户的私钥。所以急需一种可靠私钥保护协议以适应这种变化。文献[1]提出了一种能够防用户设备窃设备失窃后用户私钥的安全。但是文献[1]也指出 S－RSA 协议不能够抵抗拒绝服务攻击。文献[2]中的协议采用的密钥保护方法与文献[1]相同，也是将私钥分割。文献[2]采用双向认证的方法来抵抗拒绝服务攻击。其采取的认证方法与 Lee－Kim[3]协议非常类似。文献[4]中对 Lee－Kim 进行了详细的分析，指了了 Lee－Kim 协议存在的缺点。因此文献[2]中的协议也存在相似的缺点。

本文在对文献[1,2]协议分析的基础上，对其进行改进，提出一种 SS－RSA 协议，能够有效抵抗拒绝服务攻击，并保证用户私钥在用户设备失窃的情况下的安全。

1 S-RSA 协议分析及安全漏洞

术语定义：

① 收稿时间:2009-11-03;收到修改稿时间:2009-12-26

h( )一个安全的单向 Hash 函数。

f( )一个安全的单向函数。

⊕异或运算。

E( )加密操作

pwd 用户口令，pks 服务器公钥，pkd 用公钥，sks 服务器私钥。

S－RSA 协议可以分为三个主要的部分,1 ）用户设备初始化部分。2 ）协议正常执行部分。3 ）用户设备被盗后的用户票据撤消部分。

(1) 初始化部分：

首先向用户设备输入服务器的公钥 pks 和用户的口令 pwd 和用户公钥 pkd 。选取随机数 t,v, a,计算 u=H(t), b=H(pwd), d1=F(v,pwd), d2 = d － d1 mod ϕ (N), τ =Epks(<a,b,u,d2,N>)。其中 t, v, a, τ , pkd, pks 存储在用户设备上。其它的值从用户设备删除，而且必需保证用户可以在必要的时候离线的找回 t, τ 。因此需要将 t 和 τ 存储在其他的存储设备上。

(2) 协议正常执行部分

首先在用户设备执行如图 1 所示操作

$$\beta \leftarrow h(pwd)$$
$$\rho \leftarrow R\{0,1\}^{\lambda}$$
$$r \leftarrow R\{0,1\}^{k_{RSA}}$$
$$\gamma \leftarrow E_{pks}(<m,r,\beta,\rho>)$$
$$\delta \leftarrow mac_{\alpha}(<\gamma,\tau>)$$

图 1　用户设备执行

执行完后将 δ γ τ 发送给服务器。在这个过程中存在着拒绝服务攻击，攻击只要截获到 δ γ τ 不需要进行任何的修改，攻击者只要不断重发所截获到的信息，由于完整性和用户的口令都是正确的，就可以使得服务器不断的忙于验证所收到的信息。从而形成对服务器的拒绝服务攻击。

服务器在收到消息后执行如图 2 所示的操作。

$$<a,b,u,d_2,N> \leftarrow D_{sks}(\tau)$$
$$abort \quad if \quad mac_{\alpha}(<\gamma,\tau>) \neq \sigma$$
$$<m,r,\beta,\rho> \leftarrow D_{sks}(\gamma)$$
$$abort \quad if \quad \beta \neq b$$
$$v \leftarrow (encode(m,r))^{d_2} \bmod N$$
$$\eta \leftarrow \rho \oplus v$$

图 2　服务器收到消息后执行

服务器 d 在执行完上图的操作后将发送给用户设备。用户设备在收到之后进行如图 3 所示操作。

$$v \leftarrow \rho \oplus \eta$$
$$d_1 \leftarrow f(pwd,v)$$
$$s \leftarrow v(encode(m,r))^{d_1} \bmod N$$
$$abort \quad if \quad s^e \neq Nencode(m,r)$$
$$return <s,r>$$

图 3　用户设备收到执行

这里存在一个缺点是，由于用户设备在最后一步才对信息的完整性进行验证，如果信息通不过完整性验证，用户设备之前进行的操作全部放弃。造成用户设备计算资源的浪费。

(3) 票据销毁部分

用户在用户发现自己的设备被盗后，离线取得存储在其他设备上的 t 和 τ 然后将其发送给服务器。服务器在收到后，先将票据 τ 解密，然后验证 u?=H(t)。如果验证通过，刚将 τ 记入一个专门的废除列表里。之后对来自该票据的所有信息都不予响应。

这样做的好处是获得用户设备的攻击者无法解密用户的信息，做为一个签名协议这样做是可以的，但是做为一个加密协议来说这样做所导致的后果是一旦用户设备失窃后，由于用户的半私钥的计算要用到用户设备上的 v，当用户设备失窃后，用户也无法计算出他的半私钥，从而使得使用与被撤消私钥相对的公钥的加密的文件无法解密。

2　SS－RSA协议

本协议主要针对 S-RSA 不能抵抗拒绝服务攻击和票据取消后的恢复问题进行改进。使其能够做为一个安全的加密协议。

2.1 密钥产生阶段

由一个可信的第三方服务器，一般情况下为一个 CA，产生用户的公私钥对(N，d)和(N，e)，其中 N 为两个不同的素数 p 和 q 的乘积，e 与欧拉函数 ϕ (N)=(p－1)(q－1)互素，私钥 d 满足 ed≡1 mod ϕ (N). 然后由第三方服务器选取随机数 dc,然后计算 ds=d －dc mod ϕ (N)。最后第三服务器通过安全的信道将(e,dc, N)和(e, ds, N)传送给用户设备和服务器。第三方可信服务器按照某种托管策略对存储用户的私钥，以便于需要的时候，恢复用户的私钥。

2.2 用户初始化阶段

首先向用户设备输入服务器的公钥 pks 和用户的的公钥 pkd 以及用户的口令 pwd。用户设备执行如下操作

① 选择随机数 t，并计算 u=h(t)。用于用户设备被盗后取消票据。

② 选择随机数 v，计算 dc’=dc⊕f(v,pwd)

③ 选择随机数 a，计算 b=h(pwd)。

④ 计算 τ=(a,b,u,)。

其中 v，a，τ，pkd 和 pks 保存在用户设备上，其他的值从用户设备上删除，别外用户需要将 t 和 τ 存储在其他设备上，以便用户用户发现其设备被盗的情况下，取消票据。初始化后用户设备上存储了(a,v,τ,dc’,pkd,pks,ID,n1=1)。其中 n_1 用于服务器与用户设备同步。服务器为每一个用户存储(ID，ds，n_2=1)。

2.3 协议正常执行部分

本协议的正常执行部分分为三个阶段，其执行如图 4 所示。

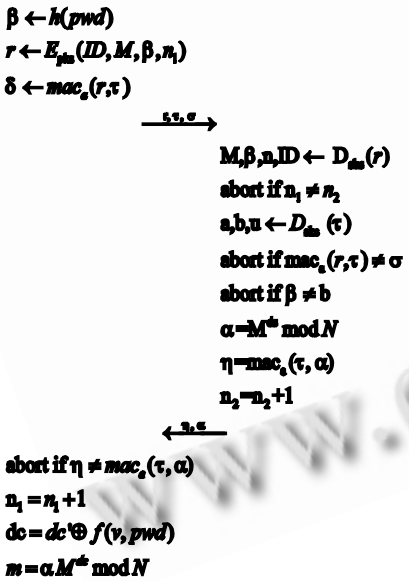

图 4 SS-RSA 执行示意图

① 如图 4 所示，用户在用户设备上输入 ID 和口令 pwd，用户设备检查 ID 是否其所存储的 ID 一致，如果不一致则停止运行并报错，如果一致则按图中所示计算 β，r，δ。其中 β 用来证明用户的合法性，r 用来保证用户数据的安全传输到的服务器，δ 用来验证用户传送数据的完整性。M 是用户要解密的数据。

② 服务器在收到用户的信息后，先将 r 解密，然后验证 n_1 与 n_2 是否相等。如果不相等则停止响应，如果相等则解密用户的票据，从中得出 a，然后验证用户信息的完整性。验证通不过，则停止响应，否则继续验证 β 是否等于 b。这里用于验证用户的合法性。因为只有合法的用户才能够生成与 b 相等的 β 值。验证通过，则通过 ID 找到服务器存储的用户半私钥 ds。使用它生成半解密文件 α。然后计算 η，将 α，η 发送给用户设备。然后将计算 n_2=n+1。以备下次验证时使用。

③ 用户设备在收到服务器传过来的 η α 后先进行完整性检查。计算 maca(τ，α)否与传过来的 η 相等，如果相等则说明是合法服务器传过来的，因为只有合法的服务器才能够解密用户票据，并且获得随机数 a。如果不等，则说明可能不是合法服务器传过来的，或者信息在传送过程中发生错误。相等则将 n_1 加 1。以便下次验证时使用。计算 dc=dc’⊕f(v,pwd)，然后与计算计算用户设备的半解密文件，最后与 η 相乘将文件完全解密。

2.4 用户证书的撤消

用户在发现自己的设备被盗或者出现其他的异常想要取消票据时，取得存储在其他设备上的 t 和用户票据 τ，计算 c=h(t)。然后将 c 与 τ 发送给服务器。服务器在收到请求后，先将 τ 解密。验证 u 与 c 是否相等，如果相等则将 τ 计入用户 ID 的废除票据列表里，对于之后来自该票据的任何请求都不响应。

3 SS－RSA协议安全性分析

将用户的 RSA 私钥分成 dC 和 dS 两部分后不会降低 RSA 算法的安全性 1，下面对 SS-RSA 协议可能遇到的各种安全威胁进行安全分析。

3.1 用户设备失窃

由于本协议是一种防用户设备失窃密钥保护协议，所以首先对这种情况进行分析。

用户设备失窃后，攻击者可以获得用户设备上的所有信息。由于用户设备上的半私钥是由用户口令和一个足够长的随机数输入一个安全的单向函数进行加密的，这就杜绝了攻击者进行离线字典攻击的可能性，所以攻击者在不知道用户口令的前提下是不能够对用户的半私钥进行离线字典攻击。所以攻击在仅仅获得用户设备的条件下，也只能对用户密码进行在线字典攻击。但是在

线字典攻击很容易被服务器端发现，并采取相应的措施。比如限制验证次数，用户在达到最大的验证次数后，在一定的时间段内对来自该用户的请求不予相应等。

3.2 用户设备与用户口令同时失窃

这种情况是最坏的情况，攻击者得到用户设备和用户口令后，完全可以冒充合法用户，要求服务器对文件进行联合解密。在这种情况下合法用户可以通过将取消码与票据发送给服务器，请求服务器停止响应来自该票据的所有请求。并将该票据记入该用户的废弃列表里。然后用户向第三方服务器请求恢复密钥，第三方服务器取出其为用户存储的密钥对用户设备和服务器进行重新初始化。

3.3 服务器被攻击

假设攻击者成功的攻击了服务器，并获得服务器上存储的所有的用户信息，那么攻击者所能做的就是跟用户设备配合，对用户文件进行解密操作。对于用户信息的危害在于配合已取消的用户票据进行解密操作，或者是拒绝对用户提供服务，造成拒绝服务攻击，但是这种情况很容易被管理员发现并采取相应的措施进行处理。最坏的情况是攻击者已获得了一个用户设备，那么它可以配合它所获得的用户设备解密用户设备上存储的信息。

3.4 口令泄露

如果用户的密码泄漏给攻击者则攻击者并不能够利用该信息进行更进一步的攻击，因为用户密码总是和存储在用户设备上的 v 值一起出现在计算中的，只要攻击者无法获得用户设备，攻击者就无法猜测 v 的值，因为我们要求 v 值是足够长的。

3.5 重放式拒绝服务攻击

由于用户设备和服务器分别保存了 n_1 和 n_2 做为计数器。当用户信息发送到服务器时，服务器首先验证 n_1 与 n_2 的值是否相等。如果相等则进行下面的操作。如果的 n_1，n_2 不相等。但是其差值在一个允许的范围之内，则继续进行信息的完整性和用户口令的合法性验证，验证通过后，则为用户提供解密服务。发送给用户的 η 变为 $maca(n_2, \alpha, \tau)$ 信息变为 n_2，α，η。用户收到后，验证信息的完整性之后，用 n_2 的值替换 n_1 的值。保证服务器与用户设备的同步。如果 n_1，n_2 的差值超过了允许的范围服务器就认为受到了攻击或者信道传输情况过差。需要采取相应的措施进行解决。由于用户设备和服务器的同步，所以本协议能够抵抗重放式的拒绝服务攻击。

3.6 用户票据恢复。

本协议采取可信第三方服务器产生用户的公私钥对，并按照某种托管协议对用户的私钥进行管理。文献[5, 6]对这方面都做了相应的研究。用户在取消自己的票据后按照托管协议的要求向可信的第三方请求重新初始化系统。这就避免了用户票据取消后用户私钥无法恢复，造成用户数据对合法用户也不可读的情况。

4 结语

本协议是对 S－RSA 协议的改进，解决了 S－RSA 不能够抵抗重放式的拒绝服务攻击和票据取消后用户私钥无法恢复的缺点。但是由于本协议也是采取的对 RSA 私钥分割处理的方法。所以不可避免的本协议也像 S－RSA 协议一样，其效率没有传统的 RSA 协议高。因为针对传统 RSA 协议的一些优化算法无法应用到 SS－RSA 协议的解密算法上。因此在目前的情况下，针对本协议的最好的用法是，对文件的加密使用对称加密算法，对于对称加密算法的密钥使用 SS-RSA 协议进行加密的混合加密算法。

参考文献

1 MacKenzie P, Reiter MK. Networked cryptographic devices resilient to capture. Proc. of the IEEE Symposium on Security and Privacy, 2001, 14－16,12.

2 虞淑瑶,叶润国,张友坤,杨宏伟.一种防窃取的私钥保存及使用方案.小型微型计算机系统, 2006,27(4): 638－641.

3 Lee SW, Kim HS, Yoo KY. Efficient nonce-based remote user authentication scheme using smart cards. Applied Mathematics and Computation, 2005,167 (1) : 355－361.

4 张利华,章丽萍,张有光,吕善伟.基于口令的远程身份认证及密钥协商协议.计算机应用,2009,29(4):924－927.

5 郭曲,何大可.对一个私钥保护协议的分析与改造.电讯技术, 2008,48(7):39－41.

6 方勇,刘嘉勇,龚海澎,等.一种私钥管理实现技术.电讯技术, 2004,44 (1):78－81.