改进的抗同步化攻击的移动双向认证协议

何 小 平
(广东培正学院数据科学与计算机学院 广东 广州 510830)

摘 要 针对汪杰等[15]设计的改进的轻量级移动 RFIO 双向认证协议进行分析,指出其协议存在的安全缺陷,在该协议基础之上,给出一种改进的能够抵抗去同步化攻击的移动双向认证协议。所提协议中,所有信息加密后再传输,且加密过程中均混入随机数,增大攻击者的破解难度。协议为能够抵抗去同步化攻击采用以下措施:在后台服务器端存放前后两次认证密钥;在标签端引入 Count 计数器。协议为降低系统整体计算量,采用位运算进行加密。对协议进行安全性分析,协议具备移动式 RFID 系统所需的安全要求;对协议进行性能分析,协议具备降低系统计算量的特征。

关键词 射频识别 移动系统 双向认证 Count 计数器 同步化攻击 随机数 交换重组交叉位运算

中图分类号 TP393.08 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2020.12.049

IMPROVED MOBILE MUTUAL AUTHENTICATION PROTOCOL AGAINST SYNCHRONIZATION ATTACK

He Xiaoping
(*School of Data and Computer Science, Guangdong Peizheng College, Guangzhou 510830, Guangdong, China*)

Abstract This paper focuses on the protocol designed by Wang et al, and it points out the security flaw of the protocol. Based on this protocol, an improved mobile mutual authentication protocol is proposed, which can resist the desynchronization attack. In the proposed protocol, all information was encrypted and retransmitted, and random numbers were mixed in the encryption process, which made it more difficult for the attacker to crack. In order to resist the desynchronization attack, the protocol adopts the following two measures: two authentication keys were stored in the back-end server; count counter was introduced in the label side. To reduce the overall calculation of the system, it used bit operation to encrypt. The security analysis of the protocol shows that the protocol has the security requirements of mobile RFID system. The performance analysis of the protocol shows that the protocol has the characteristics of reducing the system computation.

Keywords RFID Mobile system Mutual authentication Count counter Synchronization attack Random number Switching and reorganizing crossover bit operation (SRC)

0 引 言

射频识别技术不需要与实体相接触,即可识别出实体内部存放的信息,因此得到广泛应用,其中 RFID 系统的应用最为广泛[1-2]。RFID 系统在 20 世纪末得到较大范围的推广,因当时科技的局限性,使得 RFID 系统中读写器与后台服务器之间的通信都是采用有线信道完成数据传送。有线信道在很多时候被认为安全可靠,因此在设计双向认证协议之时,一般将读写器与后台服务器看成一个整体[3-5]。进入 21 世纪,随着科技的发展,人们的需求也越来越繁琐,传统的 RFID 系统已经无法满足人们的需求,急需出现新的解决方法。移动式的 RFID 系统便在这样的背景之下产生,其与传统的 RFID 系统最大的区别在于:移动式的 RFID 系统为满足人们的需求,读写器不再是固定式的,而是采

收稿日期:2019-07-08。教育部高等教育司 2018 年第二批产学合作协同育人项目(201802068001);广东高校省级重点平台和重大科研项目——特色创新项目(2018GXJK417)。**何小平**,讲师,主研领域:网络工程,信息安全。

用移动式的,即读写器与后台服务器之间的通信采用无线信道完成[6-8]。

鉴于上述移动式 RFID 系统自身特有的属性,使得适用于传统 RFID 系统的双向认证协议并不能应用在移动式 RFID 系统中,因此需要设计出新的适用于移动式的 RFID 系统的双向认证协议[9-10]。

1 相关工作

针对传统的 RFID 系统,为确保隐私信息的安全性,国内外专家学者设计出很多双向认证,以下介绍一些近些年较为经典的适用于传统的 RFID 系统的双向认证协议。

文献[11]基于 M-Hash 函数提出一种双向认证协议,但该协议仅适用于传统的 RFID 系统,无法运用在移动式的 RFID 系统中。协议因采用 Hash 函数进行加密,使得采用该协议的系统整体计算量较大,且协议无法抵抗攻击者的重放攻击。

文献[12]基于位重排变换设计出一种双向认证协议,该协议也是仅适用于传统的 RFID 系统。协议采用按位运算的加密能够一定程度上降低系统的整体计算量,对协议进行深入分析发现,协议并没有提供最后一步骤中标签与读写器之间的双向认证。

文献[13]设计出一种适用于移动式 RFID 系统的双向认证协议,协议采用伪随机函数算法对通信消息进行加密,具备一定的安全性。对协议进行分析可以发现,协议并没有文中声称的那样安全可靠,协议加密过程中,并不是所有信息均混入随机数,使得攻击者可以多次监听消息,对获取的消息进行分析,从而定位标签的位置,因此,协议无法抵抗追踪攻击。

文献[14]提出的协议能够使用在移动式的 RFID 系统中,但仍存在一定缺陷。攻击者可以采用物理攻击的手段克隆出通信实体的相关隐私信息,从而假冒成合法通信实体与其他通信实体进行通信,更进一步获取其隐私信息,因此,协议无法抵抗假冒攻击。

文献[15]提出的协议适用于移动式的 RFID 系统,对协议进行分析发现,协议在最后一步骤中,移动读写器仅仅只是转发消息给标签,从而使得协议无法提供双向认证,在此基础之上,攻击者更进一步对截获的消息进行多次重放,从而导致标签与后台服务器之间用到的共享密钥信息失去一致性,使得协议无法抵抗攻击者的去同步化攻击。

鉴于篇幅有限,此处不再阐述更多的认证协议,具体可参见文献[16-20]。

2 改进的轻量级移动 RFID 双向认证协议分析

2.1 步骤阐述

鉴于文献[15]中有详细的协议认证步骤的描述,此处为便于后续协议的具体分析,因此只对该协议认证步骤进行较为简单的描述。如图 1 所示。

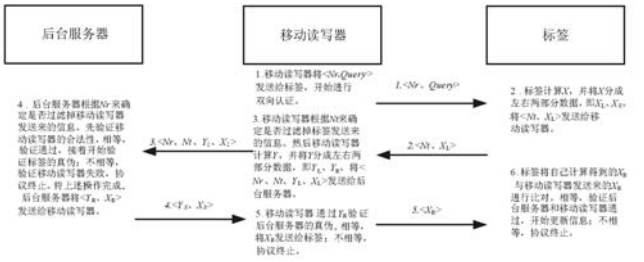

图 1 改进的轻量级移动 RFID 双向认证协议的认证步骤

认证步骤可以描述如下:

1)移动读写器产生随机数 Nr,将 Nr 和 Query 一起发送给标签。

2)标签计算通信消息 X,并将其分成左右两部分,保存右部分 X_R,将通信消息 Nt 和 X_L 发送给移动读写度。

3)移动读写器计算通信消息 Y,并将其分成左右两部分,保存右部分 Y_R,将 Nt、Nr、Y_L、X_L 发送给后台服务器。

4)后台服务器先对移动读写器进行验证,具体地,通过 Nr、Y_L 对移动读写器进行验证。验证通过,再对标签进行验证,具体通过 Nt、X_L 对标签进行验证。验证通过,开始更新信息,并将 Y_R、X_R 发送给移动读写器;验证不通过,则后台服务器会再次用到上一轮的共享密钥计算得到另一组 Nt、X_L 对标签再次进行验证。若再次验证还是未通过,协议终止;反之,标签通过验证,开始更新信息,并将 Y_R、X_R 发送给移动读写器。

5)移动读写器通过 Y_R 对后台服务器进行验证。验证通过,将 X_R 转发给标签;反之,协议终止。

6)标签通过 X_R 对后台服务器和移动读写器进行验证。验证通过,开始更新信息;反之,协议终止。

2.2 具体分析

通过上面协议的步骤描述,对协议进行深入分析,可以得到如下结论:协议无法抵抗攻击者发起的去同步化攻击。

在协议的第 5 步中,当移动读写器验证后台服务

器通过之后,就直接将 X_R 传送给标签,并没有做任何其他处理操作。

攻击者可以通过监听的手段获取第 4 步中后台服务器发送给移动读写器的消息 Y_R、X_R。当攻击者获取该消息后,立刻阻塞移动读写器与标签之间的最后一步通信。此时合法的移动读写器正在验证后台服务器的真伪,根据协议的步骤,移动读写器验证后台服务器肯定可以通过,之后移动读写器将 X_R 传送给标签。但此时合法移动读写器与标签之间已无法进行通信,攻击者就可以假冒成移动读写器向标签发送 X_R 消息。根据第 5 步可知,X_R 并不会进行任何处理,因此攻击者发送给标签 X_R 信息后,标签对移动读写器和后台服务器的验证还是可以通过,通过之后,将开始进行信息更新。此时,如果攻击者再次重放 X_R 信息给标签,则标签对其验证将会再次通过,通过之后,标签端将再次更新信息。标签端进行两次信息的更新,而此时,后台服务器端仅进行一次信息的更新,截至此刻,标签与后台服务器之间的共享密钥以及假名信息已失去一致性,攻击者去同步化攻击成功。

下面给出通过重放 X_R 信息使得标签与后台服务器之间的信息失去一致性的详细分析过程。

通信开始之前,标签与后台服务器之间信息同步,都满足 $IDt_{new} = IDt_{old} = IDt$, $Kt_{new} = Kt_{old} = Kt$,其中:IDt_{new}为当前会话的标签标识符;IDt_{old}为上一轮会话的标签标识符;IDt 为标签的标识符;Kt_{new}为当前会话的共享密钥;Kt_{old}为上一轮会话的共享密钥;Kt 为共享密钥。

第一次通信开始,通信步骤前面的第 1 步至第 5 步正常进行,攻击者阻塞第 6 步。窃听截获第 4 步中后台服务器发送给移动读写器的 Y_R、X_R 信息,在阻塞第 6 步的操作下,攻击者将 X_R 信息发送给标签,标签验证通过,开始更新信息。因这次是攻击者第一次重放 X_R 信息,因此标签端与后台服务器端的信息还是一致的。具体的更新信息如下:

$IDt_{old} = IDt_{new}$,将该次的数值记作 ID_1_old;$IDt_{new} = H_{Kt}(Nt||IDt_{old})$,将该次的数值记作 ID_1_new,其中 $H_{Kt}()$为在共享密钥 Kt 条件下的哈希函数的加密运算;$Kt_{old} = Kt$,将该次的数值记作 Kt_1_old;$Kt_{new} = H(Kt||IDt_{old})$,将该次的数值记作 Kt_1_new,其中 $H()$为哈希函数的加密运算。

当攻击者第一次重放 X_R 信息,标签端更新信息完成后,仍继续阻塞合法移动读写器与标签的通信。攻击者将进行第二次重放 X_R 信息,根据第 6 步中的描述可知,此时攻击者假冒的移动读写器还是可以再次通过标签的验证,验证通过之后,标签端将再次进行信息更新操作,但合法后台服务器却并没有进行同步的信息更新操作。截至此时,标签与后台服务器之间的共享密钥及假名已失去一致性。具体的标签端的更新信息如下:

$IDt_{new} = H_{Kt\_1\_new}(Nt||ID\_1\_new)$,将该次的数值记作 ID_2_new;$Kt_{new} = H(Kt\_1\_new||ID\_1\_new)$,将该次的数值记作 Kt_2_new。

后台服务器信息未进行更新操作,因此,此时后台服务器端存放的假名还是上面的 ID_1_old 和 ID_1_new;共享密钥还是上面的 Kt_1_old 和 Kt_1_new。

很显然,$ID\_2\_new \neq ID\_1\_old$,且 $ID\_2\_new \neq ID\_1\_new$;$Kt\_2\_new \neq Kt\_1\_old$,且 $Kt\_2\_new \neq Kt\_1\_new$。

至此,标签与后台服务器之间的共享密钥信息及假名信息均失去一致性,攻击者去同步化攻击成功。

3　抗同步化统计的协议

3.1　符号说明

抗同步化攻击的协议中出现的符号按照如下含义进行解释:

S:后台服务器;R:移动读写器;T:标签;x:移动读写器产生的随机数;y:标签产生的随机数;K:后台服务器、移动读写器、标签三者之间的共享密钥;K_{Reader}:后台服务器、移动读写器两者之间的共享密钥;K_{Tag}:后台服务器、标签两者之间的共享密钥;K_{Tag_old}:后台服务器、标签两者之间上轮认证过程中的共享密钥;ID_{Reader}:移动读写器的假名;ID_{Tag}:标签的假名;ID_{Tag_old}:标签上轮认证过程中的假名;$Count$:标签一端的计数器的数值;$SRC(X,Y)$:交换重组交叉位运算;⊕:异或运算;&:与运算。

为统一文中描述,约定用 $SRC(X,Y)$符号表示交换重组交叉位运算。交换重组交叉位运算定义如下:X 和 Y 均表示二进制序列,且位数 L 均为偶数位(如 8 位、24 位等);取二进制序列 X 的前 $L/2$ 位,放在新组成二进制序列 Z 的右边,同时取二进制序列 Y 的后 $L/2$位,放在新组成二进制序列 Z 的左边;对二进制序列 Z 进行从左到右的遍历,将二进制序列 Z 的奇数位上的数值放在新组成二进制序列的偶数位上,同时将二进制序列 Z 的偶数位上的数值放在新组成二进制序列的奇数位上。

比如:X = 1010 0001, Y = 1000 0100,根据上面的定义可得出:Z = 0100 1010, $SRC(X,Y)$ = 1000 0101。具体过程如图 2 所示。

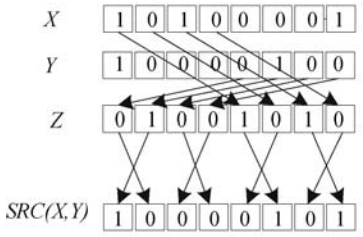

图 2 交换重组交叉位运算的流程图

3.2 协议描述

协议开始之前,后台服务器产生如下信息:ID_{Tag}、K_{Reader}、K_{Tag}、K、ID_{Reader},并通过安全通道将信息发送给标签和移动读写器。标签接收并存放下列信息:ID_{Tag}、K_{Tag}、K;移动读写器接收并存放下列信息:K_{Reader}、K、ID_{Reader};后台服务器存放下列信息:ID_{Tag}、K_{Reader}、K_{Tag}、K、ID_{Reader}。最开始的时候,令 $ID_{Tag\_old} = ID_{Tag\_new} = ID_{Tag}$;$K_{Tag\_old} = K_{Tag\_new} = K_{Tag}$。

改进的抗同步化攻击的协议过程如图 3 所示。

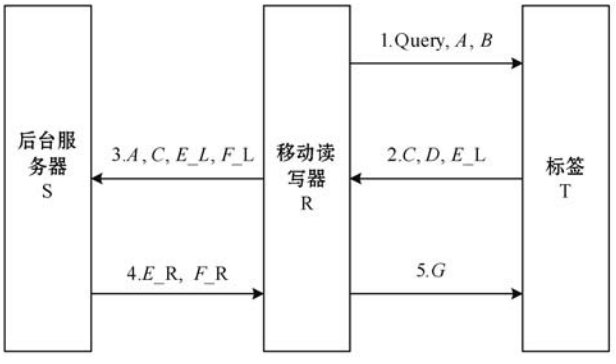

图 3 抗同步化攻击的协议

结合图 3 描述改进的抗同步化的协议步骤如下:

1)移动读写器产生随机数 x,用随机数 x、共享密钥 K 计算 $A = x \oplus K$;用随机数 x、共享密钥 K 计算 $B = SRC(x, x\&K)$。将 $\{A, B, \text{Query}\}$ 发送给标签,其中 Query 表示认证请求命令。

2)标签接收到移动读写器发送来的信息,用收到的 A、共享密钥 K 计算 $x' = A \oplus K$;用计算所得 x'、共享密钥 K 计算 $B' = SRC(x', x'\&K)$。

对比 B'与 B 是否相等。$B' \neq B$,移动读写器未通过验证,协议终止;$B' = B$,表明移动读写器通过标签的验证,同时也得出:$x' = x$。标签接着进行后续操作,产生随机数 y,用随机数 y、共享密钥 K 计算 $C = y \oplus K$;用随机数 y、计算所得随机数 x、共享密钥 K 计算 $D = SRC(y \oplus x, y\&K)$;用随机数 y、共享密钥 K_{Tag}、标签假名 ID_{Tag} 计算 $E = SRC(y \oplus K_{Tag}, y \oplus ID_{Tag})$。将消息 E 均等地分成左右两部分,左边一部分记为 E_L,右边一部分记为 E_R。标签最后将 $\{C, D, E\_L\}$ 发送给移动读写器。

3)移动读写器收到标签发送来的信息,用收到的 C、共享密钥 K 计算 $y' = C \oplus K$;用计算所得 y'、共享密钥 K 计算 $D' = SRC(y' \oplus x, y'\&K)$。

对比 D'与 D 是否相等。$D' \neq D$,标签未通过验证,协议终止;$D' = D$,表明标签通过移动读写器的验证,同时也得出:$y' = y$。移动读写器接着进行后续操作,用随机数 x、共享密钥 K_{Reade}、移动读写器假名 ID_{Reader} 计算 $F = SRC(x \oplus K_{Reader}, x \oplus ID_{Reader})$。将消息 F 均等地分成左右两部分,左边一部分记为 F_L,右边一部分记为 F_R。标签最后将 $\{A, C, E\_L, F\_L\}$ 发送给后台服务器。

4)后台服务器收到移动读写器发送来的信息,后台服务器分两步进行操作。先验证移动读写器的真伪,验证通过,再验证标签的真伪,两者均通过验证,后台服务器才会进行信息更新操作。两者中只要有一方未通过验证,协议终止。

(1)后台服务器对移动读写器的验证。后台服务器用收到的 A、共享密钥 K 计算 $x'' = A \oplus K$;用计算所得 x''、共享密钥 K 计算 $F' = SRC(x'' \oplus K_{Reader}, x'' \oplus ID_{Reader})$。将消息 F'均等地分成左右两部分,左边一部分记作 F_L',右边一部分记作 F_R'。对比 F_L'与 F_L 是否相等。$F\_L' \neq F\_L$,移动读写器未通过验证,协议终止;$F\_L' = F\_L$,表明移动读写器通过后台服务器的验证,同时也得出:$x'' = x' = x$。后台服务器接着进行后续操作,开始进行对标签的验证。

(2)后台服务器对标签的验证。后台服务器用收到的 C、共享密钥 K 计算 $y'' = C \oplus K$;用计算所得随机数 y''、共享密钥 K_{Tag}、标签假名 ID_{Tag} 计算 $E' = SRC(y'' \oplus K_{Tag}, y'' \oplus ID_{Tag})$。将消息 E'均等地分成左右两部分,左边一部分记为 E_L',右边一部分记为 E_R'。对比 E_L'与 E_L 是否相等。$E\_L' = E\_L$,表明标签通过后台服务器的验证,同时也得出:$y'' = y' = y$;后台服务器开始更新信息:$ID_{Tag\_old} = ID_{Tag\_new}$、$ID_{Tag\_new} = SRC(y, ID_{Tag\_old})$、$K_{Tag\_old} = K_{Tag\_new}$、$K_{Tag\_new} = SRC(K_{Tag}, y)$;待后台服务器更新信息完成,将 $\{E\_R, F\_R\}$ 发送给移动读写器。$E\_L' \neq E\_L$,后台服务器将取出上一轮认证过程中用到的 $< ID_{Tag\_old}, K_{Tag\_old} >$ 信息,用 $< ID_{Tag\_old}, K_{Tag\_old} >$ 替换 $< ID_{Tag}, K_{Tag} >$ 进行再次计算 $E'' = SRC(y'' \oplus K_{Tag\_old}, y'' \oplus ID_{Tag\_old})$。将消息 E''均等地分成左右两部分,左边一部分记为 E_L'',右边一部分记为 E_R''。对比 E_L''与 E_L 是否相等。$E\_L'' = E\_L$,表明标签通过后台服务器的验证,同时也得出:$y'' = y' = y$;后台服务器开始更新信息:$ID_{Tag\_old} = ID_{Tag\_new}$、$ID_{Tag\_new} = SRC(y, ID_{Tag\_old})$、$K_{Tag\_old} = K_{Tag\_new}$、$K_{Tag\_new} = SRC(K_{Tag}, y)$;待后台服务器更新信息完成,将 $\{E\_R, F\_R\}$ 发送给移动读写器。$E\_L'' \neq E\_L$,标签未通过验证,协议终止。

5）移动读写器收到后台数据库发送来的信息,将接收到的 F_R 与自身计算所得 F_R 进行比较。不相等,后台数据库未通过验证,协议终止;相等,后台数据库通过移动读写器的验证,移动读写器用收到的 E_R、自身产生随机数 x、计算所得到随机数 y 计算 $G = E\_R\&x \oplus y$。最后将{G}发送给标签。

6）标签收到移动读写器发送来的信息,首先查找与 G 相对应的计数器的数值 $Count$。$Count \neq 0$,标签不进行任何操作,直接摒弃 G;$Count = 0$,表示 G 消息是第一次发送过来,并不是攻击者重放发送来的 G 消息(以此来抵抗攻击者的去同步化攻击),并将相对应的 $Count$ 变为 1。同时标签进行下列操作:用自身计算所得的 E_R、自身产生随机数 y、计算所得到随机数 x 计算 $G' = E\_R\&x \oplus y$。

对比 G'与 G 是否相等。$G' \neq G$,移动读写器及后台服务器未通过验证,表明两者之中至少有一方是假冒的,协议终止;$G' = G$,表明移动读写器与后台服务器同时通过标签的验证,标签开始更新信息:$ID_{Tag\_new} = SRC(y, ID_{Tag\_old})$、$K_{Tag\_new} = SRC(K_{Tag}, y)$。待标签一端的信息更新完成,则双向认证到此结束。

4　协议安全性分析

4.1　双向认证

改进的协议能够提供 3 个通信实体之间的相互认证,此处选择标签与移动读写器之间的相互认证为例子进行详细分析。标签与后台服务器之间的认证、移动读写器与后台服务器之间的认证,与选择的例子分析过程相同,此处不再阐述。

在改进的协议第 2 步中,标签收到移动读写器发送来的信息之后,会对移动读写器进行第一次验证。标签先通过 A 计算得到移动读写器产生的随机数 x',再用计算得到的随机数 x'代入计算 B'中。通过比较 B'与 B 是否相等,即可识别出移动读写器的真伪。具体的识别验证过程,可以参见协议的第 2 步描述。

在协议的第 3 步中,移动读写器收到标签发送来的信息之后,对标签的真伪进行辨识。通过 C 计算得到标签产生的随机数 y',将计算得到的随机数 y'代入 D 的计算,可以得到 D'。通过比较 D'与 D 是否相等,即可辨别出标签的真伪。

此处重点分析协议步骤中的最后一步中,标签是如何通过 G 同时识别出移动读写器和后台服务器的真伪。从 $G = E\_R\&x \oplus y$ 公式中可以看出,G 在计算过程中包含 3 个参数:E_R、x、y;这 3 个参数对于攻击者来说,除了 E_R 可以通过窃听手段获取之外,x 和 y 是攻击者无法获取的,且也无法通过破解获得。因此,当攻击者窃听得到 E_R 之后,伪装成移动读写器进行加密计算,但因缺少 x 和 y 两个参数,是无法计算得到正确的 G。标签收到信息后,通过简单的计算,即可发现 G 不正确,协议终止,表明移动读写器与后台服务器之间至少有一方是伪造的。

基于以上阐述,改进的协议能够提供通信实体之间的双向认证。

4.2　假冒攻击

假冒攻击是指攻击者伪装成合法的通信实体中的任何一方,向其他通信实体发送消息,以达到蒙混过关的目的,从而进行隐私信息的窃取。

改进的协议能够抵抗攻击者的假冒攻击。具体来说,协议在进行所有合法操作之前,均会对消息的来源一方进行识别认证,只有在验证消息来源一方不是伪造的之后,才会进行后续操作。攻击者想假冒成其中一个通信实体,则在进行消息加密过程中,会用到一些参数,比如共享密钥 K。对于这些参数的具体数值,攻击者是无法获取的,因此攻击者在加密过程中,就只能随便选择一个数来进行消息的加密,很明显所得结果肯定是错误的。当攻击者加密的消息发送给其他通信实体的时候,另一个通信实体就只需要进行简单的计算比对,即可识别出消息的来源一方是伪造的,攻击者的假冒攻击失败。因此,协议能够抵抗攻击者的假冒攻击。

4.3　重放攻击

重放攻击是说攻击者在通过监听手段获取某些通信消息之后,伪装成合法的一个通信实体,重复多次给另外的通信实体发送该截获的通信消息,从而达到某种目的,比如获取隐私信息。

改进的协议中为抵抗攻击者的重放攻击,所有通信消息在加密过程中,均混入随机数;其中有些随机数是移动读写器产生的,有些随机数是标签产生的;随机数的混入,既可以保证前后两轮通信过程中的通信消息的互异性,也可以增加攻击者的破解难度。当攻击者通过监听等手段获取当前的通信消息,想通过重放该消息以逃过另一通信实体的认证,但攻击者无法成功,因为下一轮认证过程中用到的通信消息加密过程中的随机数已经发生变化,所以攻击者重放该消息失败。基于以上阐述,协议能够抵抗攻击者的重放攻击。

4.4　去同步化攻击

去同步化攻击也称为异步攻击,主要是攻击者通过重放消息或其他手段,导致通信实体双向认证用到

的共享密钥等信息失去一致性,从而使得后续正常的认证无法进行。

改进的协议为了能够抵抗攻击者的去同步化攻击,采用以下两个措施:(1) 在后台服务器一端不仅存放当前认证过程中用到的共享密钥等信息,而且也会存放上一轮认证过程中用到的共享密钥等信息,这样就可以使得后台服务器对标签的认证第一次失败之后,还可以用上轮信息替换本轮信息再次认证,从而恢复同步性。(2) 标签一端会增加计算器参数 *Count*,通过该计算器的数值,来进行相对应的操作。当 *Count* 不为零时,表明标签至少已经在此次通信之前接收到通信消息 G 一次,为避免攻击者发起的异步攻击,标签采取放弃策略,即不进行信息的更新操作,从而保持标签与后台服务器之间的共享密钥等信息是一致的。而当 *Count* 为零时,表明标签之前还未收到该信息 G,从而可以断定该信息并不是攻击者重放的,标签开始对 G 进行验证。验证通过,开始更新信息;验证失败,协议终止。通过上述两个措施,使得改进的协议能够抵抗攻击者发起的去同步化攻击 。

4.5 追踪攻击

攻击者通过多次监听通信实体发送的消息,根据获取的通信消息分析出具体的通信实体,从而进行跟踪,然后实施破坏等操作。为能够抵抗攻击者发起的追踪攻击,改进的协议在加密过程中混入随机数,使得攻击者无法实施追踪攻击。主要原因有下面几点:(1) 随机数由随机数产生器随机产生,无法进行预测;(2) 随机数在一个时间段内具有不重复性;(3) 消息加密过程中用到的随机数每轮均不一样,即便是获取当前的通信消息,也无法推导出下轮通信消息。基于上述因素,使得攻击者发起的追踪攻击失败。

4.6 暴力破解攻击

当攻击者无法通过上述等攻击方法获取隐私信息之时,攻击者可能会采取更为直接更为暴力的方式进行,即采用穷举的方式穷尽所有可能的情况,然后获取隐私信息。改进的协议能够抵抗攻击者的暴力破解攻击,具体此处以通信消息 A 为例子进行详细分析。在通信消息 A 中 $A = x \oplus K$ 包含 2 个加密参数:x 和 K,攻击者对 A 无法实施暴力破解攻击获取隐私信息。具体原因如下:(1) 对于攻击者来说,这 2 个参数都是不知晓的,且无法通过监听等手段获取;(2) 共享密钥 K 在通信过程中没有明文出现过,攻击者要想获取 K,还必须采用其他手段获取;(3) 随机数 x 每轮均发生变化,无法预测,使得通信消息 A 的值也是每轮不同。基于上述,协议能够抵抗攻击者发起的暴力破解攻击。

本文协议与其他此类移动双向认证协议进行安全性分析对比,对比的结果如表 1 所示。

表 1 认证协议安全性分析对比

攻击类型	文献[11]	文献[12]	文献[13]	文献[14]	文献[15]	本文协议
适用移动 RFID 系统	×	×	√	√	√	√
双向认证	√	×	√	√	√	√
重放攻击	×	√	√	√	√	√
异步攻击	√	√	√	√	×	√
追踪攻击	√	√	×	√	√	√
暴力破解	√	√	√	√	√	√
假冒攻击	√	√	√	×	×	√

注:√表示能够抵抗或适用,×表示无法抵抗或不适用。

5 性能分析

移动式的 RFID 系统中一般包含三大通信实体,即移动读写器、后台服务器、标签。在三大通信实体中,后台服务器与移动读写器两者具备强大的计算能力和足够大的存储空间,因此性能分析部分一般不会选择这两者作为研究对象。相反,标签并不具备前面两者的优势,计算能力非常有限,且存储空间大小也受到严格限制,无法大量存放数据,因此性能分析中会将标签作为重点研究对象进行分析。同时将会选择标签的计算量和存储量作为两个性能分析的角度。

本文协议与其他认证协议进行性能分析比对结果如表 2 所示。

表 2 认证协议性能分析比对

比较对象	计算量	存储量
文献[11]	$2H + PR$	4l
文献[12]	$11M + 9Reg + 9Rot$	5l
文献[13]	$3F + PR$	3l
文献[14]	$F + C$	3l
文献[15]	$3F + 2M + 2PUF$	4l
本文协议	$3XOR + 5SRC + AND + PR$	4l

表 2 中:H 表示哈希函数的计算量;M 表示模运算的计算量;Reg 表示位重排变换的计算量;Rot 表示左循环移位的计算量;F 表示伪随机函数的计算量;PR 表示随机数产生的计算量;C 表示交叉运算的计算量;PUF 表示物理不可克隆函数的计算量;XOR 表示异或运算的计算量;AND 表示与运算的计算量;SRC 表示交换重组交叉位运算的计算量。在上述计算量中,属

于超轻量级的计算有:Reg、Rot、C、XOR、AND、SRC;其他计算属于轻量级的计算量。轻量级的计算量一次操作的计算量相当于若干次超轻量级计算的计算量。

本文协议在第 2 步中,计算随机数 x' 时第一次用到 XOR 运算,计算 C 时第 2 次用到 XOR 运算,计算 B'、D、E 时会连续 3 次用到 SRC 运算,产生随机数 y 用到一次 PR 运算。在协议的第 6 步中,计算 G' 时,第一次用到 AND 运算,同时也是第 3 次用到 XOR 运算,共享密钥和假名在更新过程中会连续 2 次用到 SRC 运算。基于上述,本协议在认证过程中,标签一端的计算量为:$3XOR + 5SRC + AND + PR$。其他协议标签一端的计算量按照上述方法可以逐一查找出来,鉴于文中篇幅有限,此处不再具体阐述。

设定所有消息的长度均为 l 位,本协议标签一端会存放如下信息:共享密钥 K、假名 ID_{Tag}、共享密钥 K_{Tag}、计数器 $Count$。因此,协议标签一端的存储量为 4l。

结合表 2 可以得出:本文协议标签一端的计算量相对于其他协议来说,得到一定程度上的减少。从标签一端的存储量角度出发,本文协议与其他协议相当,本文协议在此方面并没有较大的改进。综合各方面比较,本文协议在计算量方面要优于其他协议,且本文协议能够弥补其他协议存在的一些安全缺陷问题,具备一定的运用价值。

6　结　语

本文对汪杰[15]等提出的移动认证协议进行深入分析,指出该协议存在的安全不足,并给出一种改进的能够抵抗去同步化攻击的移动双向认证协议。改进协议为抵抗攻击者的去同步化攻击,在后台服务器存放上一轮和本轮的认证共享密钥信息,且在标签引入计数器 Count,根据计数器 Count 的数值进行不同的操作。对协议进行安全性分析,改进的协议能够满足移动式 RFID 系统的安全需求;对协议进行性能分析,表明改进的协议具备计算量低等特征。下一步研究方向是将该协议在具体的移动式 RFID 系统中实现,对一个完整的通信过程进行分析,研究分析一个完整通信过程所需时间等参数。

参　考　文　献

[1] 刘鹏. 一种 RFID 系统多标签共存证明协议设计[J]. 兵器装备工程学报, 2018, 39(2): 124 - 126.

[2] Xie R, Jian B Y, Liu D W. An improved ownership transfer for RFID protocol[J]. International Journal of Network Security, 2018, 20(1): 149 - 156.

[3] Gao L, Zhang L, Lin F, et al. Secure RFID authentication schemes based on security analysis and improvements of the USI protocol[J]. IEEE Access, 2019, 7: 8376 - 8384.

[4] Khadka G, Bibile M A, Arjomandi L M, et al. Analysis of artifacts on chipless RFID backscatter tag signals for real world implementation [J]. IEEE Access, 2019, 7: 66821 - 66831.

[5] Shen L, Zhang Q, Pang J, et al. PRDL: Relative localization method of RFID tags via phase and RSSI based on deep learning[J]. IEEE Access, 2019, 7: 20249 - 20261.

[6] 刘道微, 凌捷, 杨昕. 一种改进的满足后向隐私的 RFID 认证协议[J]. 计算机科学, 2016, 43(8): 128 - 130, 158.

[7] Li B, He Y, Liu W, et al. Towards time efficient localized polling for large-scale RFID systems [J]. Computer Networks, 2019, 150(26): 250 - 262.

[8] Parchin N O, Basherlou H J, Abd-Alhameed R A, et al. Dual-Band monopole antenna for RFID applications[J]. Future Internet, 2019, 11(2): 31.

[9] Jia X, Bolic M, Feng Y, et al. An efficient dynamic anti-collision protocol for mobile RFID tags identification [J]. IEEE Communications Letters, 2019, 23(4): 620 - 623.

[10] Xie R, Ling J, Liu D. Wireless key generation algorithm for RFID system based on bit operation[J]. International Journal of Network Security, 2018, 20(5): 938 - 949.

[11] 张兴, 李畅, 韩冬, 等. 基于 Hash 轻量级的 RFID 安全认证协议[J]. 计算机工程与设计, 2018, 39(5): 1269 - 1275, 1309.

[12] 黄可可, 刘亚丽, 殷新春. 基于位重排变换的超轻量级 RFID 双向认证协议[J]. 计算机应用, 2019, 39(1): 118 - 125.

[13] Huang Z, Xu R, Chu C, et al. A novel cross layer anti-collision algorithm for slotted ALOHA-based UHF RFID systems [J]. IEEE Access, 2019, 7: 36207 - 36217.

[14] Fan K, Jiang W, Li H, et al. Lightweight RFID protocol for medical privacy protection in IoT [J]. IEEE Transactions on Industrial Informatics, 2018, 14(4): 1656 - 1665.

[15] 汪杰, 汪学明. 改进的轻量级移动 RFID 双向认证协议[J]. 计算机工程与设计, 2018, 39(4): 912 - 917.

[16] Aghili S F, Mala H. Security analysis of an ultra-lightweight RFID authentication protocol for m-commerce[J]. International Journal of Communication Systems, 2019, 32(3): e3837.

[17] Xu H, Shen W, Li P, et al. Novel implementation of defence strategy of relay attack based on cloud in RFID systems [J]. International Journal of Information and Computer Security, 2019, 11(2): 120 - 144.

(下转第 327 页)

5 结　语

为检测开源软件中存在的大量漏洞,本文提出了一种基于双向 LSTM 的 Java 漏洞检测方法。首先运用静态分析提取源代码语义特征并生成中间表示,然后在将生成的中间表示映射为向量的同时为其贴上“是否安全”的标签,最后运用神经网络在数据集上训练生成漏洞检测模型。静态分析主要包括抽象语法树、数据流和控制流分析,目的是提取方法完整的数据和控制依赖,以提高漏洞检测的准确性。神经网络选择双向 LSTM,目的是可以更好地学习源代码的序列信息。

实验结果表明,模型在测试集上取得了 93.8% 的准确率和 90.1% 的召回率,均优于现有的基于机器学习的漏洞检测方法,验证了本文方法的优越性。后续工作主要分为两个方面,一是进一步提升模型的精确度,减少人工参与,实现开源软件漏洞的自动化检测;二是尝试将本文方法扩展到除 Java 外的其他语言,验证其适用性。

参 考 文 献

[1] Snyk. The state of open source security report[R/OL]. [2019-07-02]. https://bit.ly/SoOSS2019.

[2] Shankar U, Talwar K, Foster J S, et al. Detecting format string vulnerabilities with type qualifiers[C]//Proceedings of the 10th conference on USENIX Security Symposium. ACM, 2001, 10: 16.

[3] Yamaguchi F, Golde N, Arp D, et al. Modeling and discovering vulnerabilities with code property graphs[C]//2014 IEEE Symposium on Security and Privacy. IEEE, 2014: 590 - 604.

[4] Qian C X, Luo X P, Le Y, et al. VulHunter: Toward discovering vulnerabilities in android applications[J]. IEEE Micro, 2015, 35(1): 44 - 53.

[5] Eschweiler S, Yakdan K, Gerhards-Padilla E. DiscovRE: Efficient cross-architecture identification of bugs in binary code[C]//The Network and Distributed System Security Symposium(NDSS), 2016.

[6] Li Y K, Chen B H, Chandramohan M, et al. Steelix: Program-state based binary fuzzing[C]//2017 11th Joint Meeting on Foundations of Software Engineering. ACM, 2017: 627 - 637.

[7] Wang J J, Chen B H, Wei L, et al. Skyfire: Data-driven seed generation for fuzzing[C]//2017 IEEE Symposium on Security and Privacy (SP). IEEE, 2017: 579 - 594.

[8] Blanda A. Fuzzing Android: a recipe for uncovering vulnerabilities inside system components in Android[C]//Black Hat Europe, 2015.

[9] Shin Y, Williams L. Can traditional fault prediction models be used for vulnerability prediction? [J]. Empirical Software Engineering, 2013, 18(1): 25 - 59.

[10] Pan X R, Wang X Q, Duan Y, et al. Dark hazard: Learning-based, large-scale discovery of hidden sensitive operations in android apps[C]//Network and Distributed System Security Symposium, 2017.

[11] Chowdhury I, Zulkernine M. Using complexity, coupling, and cohesion metrics as early indicators of vulnerabilities[J]. Journal of Systems Architecture, 2011, 57(3): 294 - 313.

[12] The MITRE Corporation. Common weakness enumeration? [DB/OL]. [2019-07-02]. https://cwe.mitre.org.

[13] Parr T, Harwell S, Vergnaud E, et al. ANTLR4[CP/OL]. [2019-07-02]. https://github.com/antlr/antlr4.

[14] Arzt S, olhotak, mbenz89, et al. Soot[CP/OL]. [2019-07-02]. https://github.com/Sable/soot.

[15] National Institute of Standards and Technology. NIST software assurance reference dataset[DS/OL]. [2019-07-02]. https://samate.nist.gov/SARD.

[16] Li Z, Zou D Q, Xu S H, et al. VulDeePecker: A deep learning-based system for vulnerability detection[C]//The Network and Distributed System Security Symposium, 2018.

[17] 李元诚,黄戎,来风刚,等. 基于深度聚类的开源软件漏洞检测方法[J]. 计算机应用研究, 2020, 37(4): 1107 - 1110, 1114.

(上接第 308 页)

[29] 李晓峰,赵海,王家亮,等. 基于增加一个随机数的 ElGamal 数字签名算法的改进[J]. 东北大学学报(自然科学版), 2010, 31(8): 1102 - 1104, 1112.

[30] 喻秋叶. 生日悖论在密码学中的应用[D]. 武汉:华中师范大学, 2013.

[31] 张先红. 数字签名原理及技术[M]. 北京:机械工业出版社, 2004: 95 - 95.

[32] 韩益亮,杨晓元,户军茹,等. 改进的 ECDSA 签名算法[C]//第二十届全国数据库学术会议论文集(研究报告篇), 2003.

(上接第 315 页)

[18] Lin Q, Yang L, Guo Y. Proactive batch authentication: Fishing counterfeit RFID tags in muddy waters[J]. IEEE Internet of Things Journal, 2019, 6(1): 568 - 579.

[19] Rashid N, Choudhury S, Salomaa K. Localized algorithms for redundant readers elimination in RFID networks[J]. International Journal of Parallel, Emergent and Distributed Systems, 2019, 34(3): 260 - 271.

[20] Corchia L, Monti G, Tarricone L. A frequency signature RFID chipless tag for wearable applications[J]. Sensors, 2019, 19(3): 494.