一种基于散列链的无线网络重编程安全认证机制

任 超 张 羽 方智毅*
西北工业大学计算机学院, 西安 710072

摘要 现有的无线传感器网络重编程安全机制大多基于无线重编程协议Deluge, 主要解决无线重编程中更新代码的安全认证问题, 但恶意节点很容易通过伪造控制消息对参与无线重编程的节点实施DoS攻击. 文中将"抗DoS方案ADV-Hash"与"更新代码安全认证协议"相结合, 将广告散列链和更新代码散列链相结合, 提出了一种基于散列链的无线传感器网络重编程安全认证的改进方案ADV-Data-Hash. 该方案利用散列链和一次数字签名相结合的方案实现了对无线重编程轻量认证. 理论分析和仿真实验表明该方案在不影响Deluge协议原有特性的同时, 能有效避免由虚假广告引发的DoS攻击以及对更新代码的篡改和伪造, 确保网络重编程协议的可用性.

关键词 无线传感器网络 重编程 散列链 安全认证

无线传感器网络(wireless sensor networks, WSNs)在环境恶劣或无人可达的地区执行任务时, 经常会产生部署后改变传感器节点行为的需要, 需要改变传感器节点上当前运行的程序, 为其添加或删除指定的功能[1,2]. 无线重编程(reprogramming)通过无线链路将更新代码传播到整个网络, 是改变和完善无线传感器网络节点功能的有效途径[3,4].

Deluge[6]是目前占主导地位的无线重编程协议, 它提供了可靠有效的代码分发机制, 但没有考虑分发过程中的安全性问题. 目前基于Deluge的常见安全解决方案有Sluice[5], SecureDeluge[6], Deng-Tree[7], Deng-hybrid[7]以及SCPK[8]等, 均采用了基于公钥密码的数字签名方案, 给Deluge协议增加了安全机制. 但是由于Deluge协议本身的三次握手和"流行病"特性使攻击者很容易利用虚假控制消息进行DoS攻击[9], 从而严重降低了无线重编程服务的可用性.

为解决虚假广告引起的DoS攻击, 张羽等提出了ADV-Hash方法[1)], 其利用Deluge协议广告消息(Advertisement, Adv)的有限性和有序性, 采用基于单向散列链的认证机制, 较好地解决了虚假广告DoS攻击问题. 本文对ADV-Hash方法和更新代码认证方案做出改进, 并将两者进行结合, 提出了一种基于散列链的无线重编程安全认证机制. 本方案对虚假广告引起的DoS攻击有较强的抵抗能力, 同时没有改变Deluge协议的"流行病"特性, 在更新代码包(Data package)的认证过程中能够容忍包的乱序. 仿真实验表明, 本方案在较少的增加计算、存储和通信开销的前提下较好地实现了上述安全性需求.

1 ADV-Hash简介

ADV-Hash方法通过对广告消息进行轻量认证来解决虚假广告DoS攻击问题, 其主要思路如下.

1.1 基本思想

我们使用如下符号系统来ADV-Hash.

2009-03-26收稿, 2009-06-09收修改稿

* 通信作者, E-mail: moren@263.net

1) 方智毅, 王丽芳, 张 羽, 等. 无线传感器网络重编程中DoS攻击研究. 西北工业大学学报. 出版中

表1 ADV-Hash 使用的符号系统

符号	含义
S	可信基站
X	用户需更新的程序映像
M_t	类型为 t 的消息(t: Adv, Req, $Data$)
PK	公钥体制中的一个公钥
SK	公钥系统中的一个私钥
$[X]_{SK}$ 或 $SIGN(X)$	对 X 做数字签名
$H(X)$ 或 $HASH(X)$	对 X 做散列运算
//	连接操作
$N_{[i]}$	距离基站跳数为 i 跳的节点
p	程序映像被分页后的总页数

ADV-Hash 将更新代码镜像像所对应的 M_{adv} 看成一个有限的广告序列 $Seq: M_{adv}$. 假设镜像 X 的大小为 p 页, 则在一次重编程过程中, 基站发出的合法广告 M_{adv} 共有 $p+1$ 种, 其序列如下:

$$Seq: M_{adv} = \{(v', i) \mid i = 0, 1, 2, \cdots, p\}$$

其中 v' 是更新程序映像的版本号. 基站在分发代码前, 首先构建广告序列 $Seq: M_{adv}$ 散列链, 将最后一个广告 $Adv(v', p)$ 和填充该包的 0 进行散列运算得到 H_p, 把其存入前一个广告 $Adv(v', p-1)$ 中, 再对这个广告及 H_p 做散列运算, 并逐步计算得到 H_0, 即 $H((v', 0) \| H_1)$), 最后将 H_0 做一次数字签名得 $[H_0]_{SK}$, 并使其与 H_0 一起作为基站的第一个分发包. 如图 1 所示:

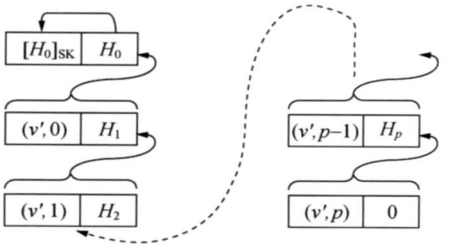

图 1 广告散列链构建示意图

无线重编程开始前, 基站将 $Seq: M_{adv}$ 序列中的广告广播给第一跳的节点 $N_{[1]}$, $N_{[1]}$ 先用已预先保存在其上的基站公钥 PK 对广告散列上的链首做签名验证, 再用 Hash 函数对后续散列逐一验证. 在接受基站所发广告后, 根据 $N_{[1]}$ 自身拥有更新程序映像代码页的情况来向其邻居节点 $N_{[2]}$ 广播相应的 Adv. 每个 $N_{[i]}$ 节点在听到一个 Adv 时, 将与 $N_{[1]}$ 一样做相同的操作来验证该 Adv 的合法性.

ADV-Hash 方案限制了节点自己生成 Adv, 所有的节点广告都源自基站, 并由其签名, 节点自己不能随意生成或修改. 同时, ADV-Hash 方案中的 Adv 也同样利用 Deluge 的“流水线”技术, 到达快速并行分发的效果.

1.2 认证算法

ADV-Hash 方法分成两个部分. 一部分在基站上完成. 由于安全重编程研究一般假设基站是绝对安全的, 因此可信基站负责将更新代码镜像中各页的广告构建如图 1 所示的 Adv 散列链和 Adv 链首的数字签名.

另一部分则在 WSNs 每个传感器节点上进行, 每个传感器节点验证所接收的 Adv, 若通过则触发三次握手过程进行相应代码页的传输.

1.3 ADV-Hash 方法分析

ADV-Hash 方法不允许节点自己生成广告, 所有的广告均由基站产生, 在一次签名后发放, 因此 ADV-Hash 认证方法可以抵御虚假广告 DoS 攻击, 确保网络重编程协议的可用性. 此外, 由于广告的签名是在可信基站上完成, 节点只保存数字签名的对应公钥, 因此即使节点被捕获, 攻击者也无法伪造广告散列链.

ADV-Hash 提供了良好的 Adv 认证方法, 但无线重编程中的对更新代码进行认证时, 也需要进行一次数字签名验证计算和若干次 Hash 运算[5−7], 加重了节点在无线重编程过程中的计算、存储和通信开销. 尤其是数字签名验证计算的计算量对于 WSNs 节点来说是比较大的, 因此需要进行改进.

2 改进的方案 ADV-Data-Hash

由于 ADV-Hash 方案只提供了广告的安全认证而没有对更新代码进行认证, 而现有的更新代码认证方案存在一些问题(见表 2), 本节将用新的更新代码认证方案, 并与 ADV-Hash 整合, 提出改进的基于 Hash 散列的无线重编程安全认证方案.

表 2 现有安全认证方案的缺点

方案	缺点描述
Sluice	验证粒度太大(页级), 一个错误包将导致整个页重传.
SecureDeluge	Deluge 协议是页有序但包不一定有序, 因此建立在包级粒度的单向 Hash 链不能容忍包乱序.
Deng-tree	存储开销较大, 实现困难.
Deng-hybrid	

2.1 方案简介

对更新代码采用包级的验证, 对于第 i 页代码, 将各包的散列值 $H_{i,j}$ 存入第 i-1 页对应的包 $D_{i-1,j}$ 中; 第 0 页的散列值作为单独的一列, 等待进一步处理. 如图 2 所示.

当包 D_{ij} 到达时, 由于第 i-1 页肯定已经全部到达, 因此该包的验证码 H_{ij} 已经随着包 $D_{i-1,j}$ 到达.

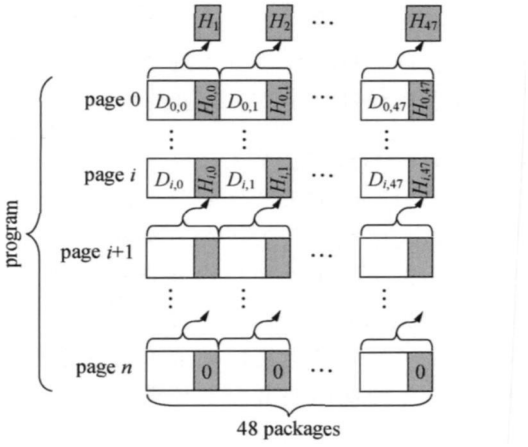

图 2 构建 **Data-Hash** 示意图

对广告消息, 借用 ADV-Hash, 把刚才第 0 列的散列值加入到 ADV-Hash 散列中. 首先, 将 H_1 至 H_{47} 共 48 个 Hash 值按每 7 个一组, 组装在一起, 如图 3 所示:

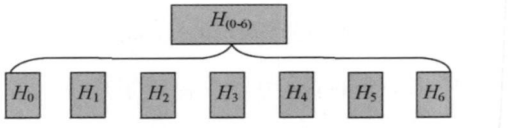

图 3 **Hash** 值的组合示意图

最后生成如图 4 所示的 ADV-Hash 链, 将更新代码的 Hash 值连接其中.

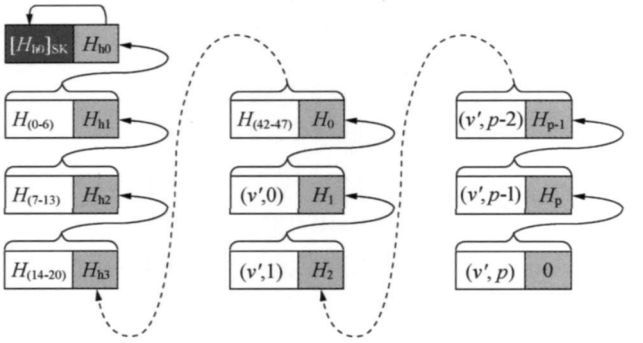

图 4 **ADV-Data-Hash** 链示意图

2.2 认证过程

在进行代码分发前, 由基站生成更新代码 Hash 链(见图 2), 再将第 0 列 Hash 值组成 $H_{(i\sim i+6)}$, $i\in\{j \mid j\%7=0,\ 0\leqslant j\leqslant 42\}$, 最后构建 ADV-Data-Hash 链, 见图 4.

在重编程过程中, 基站首先广播 ADV-Data-Hash 链, 节点将对收到的 ADV 以及更新代码进行认证. 处理过程如图 5、图 6 所示:

```
Partition X into p pages
    p1, p2, …, pp
Construction Data-Hash
    for all (i = p to i = 1) do
      for all (j = N−1 to j = 0) do
        h_{i,j} ← HASH (M_{data,i,j})
        M_{data,i−1,j} ← (M_{adv,i−1} ‖ h_{i,j})
      end for
    end for
Construction { H_{(i−i+6)} | i%7=0, 0≤i≤42}
Construction ADV-Data-Hash
    for all (i = p to i = 2) do
      h_i ← HASH (M_{adv,i})
      M_{adv,i−1} ← (M_{adv,i−1} ‖ h_i)
    end for
    h_0 ← HASH (M_{adv,1})
    M_{H(42−47)} ← (M_{H(42~47)} ‖ h_0)
    for all (i = 7 to i = 1) do
      h_hi ← HASH (M_{H(6i~6i+6)})
      M_{H(6i~6i+6)} ← (M_{(6i~6i+6)} ‖ h_hi)
    end for
    σ ← SIGN (H_h0)
    H_h0 ← (σ ‖ H_h0)
Broadcast ADV−Data−Hash
```

图 5 可信基站算法处理过程

```
Static h, i = 0
while i < p+1+7 do
  N[i+1] complete receiving ( [H_h0]_SK || H_h0 )
  if (i = 0) then
    σ, PK ← ( [H_h0]_SK || H_h0 )
    if Verify (σ, PK) then
      Accept ( [H_h0]_SK || H_h0 )
    end if
  else
    if (i< 7) then
      h' ← HASH (M_hi)
      if (h = h') then Accept (M_hi)
      end if
  else
      h' ← HASH (M_adv,i-1)
      if (h = h') then
        Accept (M_adv,i-1)
      end if
procedure Accept (M_adv,i-1)
    broadcast M_adv,i-1 according to available update
    save M_adv,i-1 in local RAM
    if (i< p) then
      h ← h_i
      i ← i + 1
    end if
end procedure
```

图 6 传感器节点算法处理过程

2.3 性能分析

(1) 计算开销：假设一个更新程序映像包含 P 页，每页包含 N 个包($N=48$)．在 Deluge 中整个传输数据与控制消息的比例为 $\Delta=81.92:18.18$[4]．因此控制消息的总量为 ΔP．在无 DoS 攻击的情况下，每个传感器节点利用该认证方法的计算开销为：

$$T_{Snd}=k\Delta P\cdot T_{Sig}+(NP+7)\cdot T_{Hash}$$

$$T_{Rec}=k\Delta P\cdot(P\cdot T_{Hash}+T_{Veri})+(NP+7)\cdot T_{Hash}$$

$$T=T_{Snd}+T_{Rec}=k\Delta P\cdot(T_{Sig}+P\cdot T_{Hash}+T_{Veri})+2(NP+7)\cdot T_{Hash}$$

其中 T_{Veri} 是节点(接收者)数字签名认证所需的计算时间，T_{Hash} 代表节点计算一个散列值所需的时间，T_{Sig} 是基站(发送者)数字签名时间，$k=10.26/18.18$[4] 是广告消息占控制消息的比例．

(2) 存储开销：此处的存储开销是指传感器节点利用该方法而需要额外使用的存储空间大小．在 ADV-Hash 方法基础上，该方案增加了额外的 64 B 存储空间用来存储 H_0 至 H_{47}，即总共占用传感器节点 RAM 空间 634 B + 64 B，占用 ROM 空间 3.5 KB.

(3) 通信开销：该方法使 WSNs 更新一个程序映像需要增加广告的额外传输开销为 $S_{Sig}+(P+1+7)\cdot S_{hash}+NP\cdot S_{hash}$．其中 S_{Sig} 是广告散列链链首中一个数字签名的大小，S_{hash} 是一个广告中所含散列值的大小．

3 仿真实验

本节对认证方法进行了 TOSSIM[10] 仿真，并将 WSNs 重编程协议 Deluge 和已有的安全网络重编程协议 Sluice 以及 ADV-Hash 作为实验比对协议．以下所有仿真实验中，参数设置为：$\tau_l=2$ s，$\tau_h=60$ s，$k=2$，$\tau_r=0.5$，$\lambda=2$，$\omega=8$．实验 1 显示的是在无 DoS 攻击时，三种协议“完成时间”的比较，如图 7 所示：

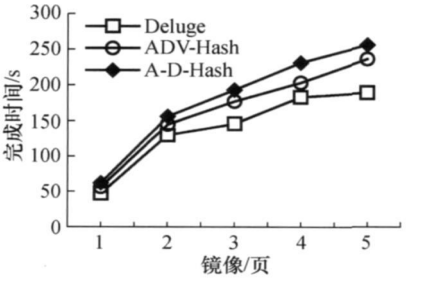

图 7 无 **DoS** 攻击情况下重编程完成时间比较

图 7 表示在 4×4、节点间距离为 15 ft(1 ft=0.3048 m)的网格中，Deluge、ADV-Hash 和本文提出的 ADV-Data-Hash 方案(在图 8、图 9 中简称“A-D-HASH”方案)的网络重编程完成时间比较情况．Deluge 没有安全机制，因此在无 DoS 攻击时完成时间最短．ADV-Data-Hash 安全机制的开销略大于 ADV-Hash 方案，因为 ADV-Data-Hash 比 ADV-Hash 增加了对更新代码的认证．

实验 2(图 8)为本文认证方案与 ADV-Hash、Sluice 在受到虚假更新代码攻击时的完成时间比较．

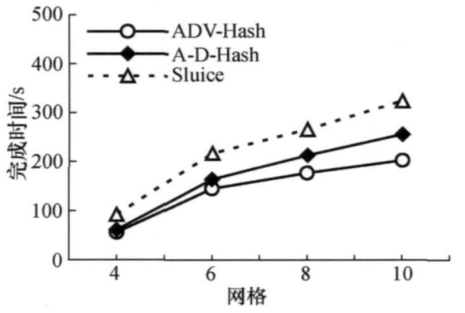

图 **8** 虚假更新代码攻击

图8显示在受虚假更新代码攻击时，不同规模下ADV-Hash，Sluice与本文认证方案的网络完成时间比较. 该图反映了在 $N\times N$(N=4，6，8，10)的WSNs(节点间距离15ft)中进行大小为1页的程序镜像网络重编程并受到虚假更新代码攻击时，A-D-Hash方案由于采用了轻量级认证，因此完成时间小于Sluice. Sluice在对更新代码进行认证时，采用了数字签名验证算法，而ADV-Data-Hash方案在对更新代码进行认证时采用Hash运算，在被攻击时受到的影响远小于Sluice.

图中虽然显示ADV-Hash完成时间最短，但其不能认证更新代码的合法性，不能达到安全重编程的要求.

4 总结

本文分析了现有的无线重编程安全认证方案的特性以及存在的缺陷，改进了更新代码认证方案，并将其与ADV-Hash方法相结合，提出了ADV-Data-Hash方案. 该方案用较小的开销(一次数字签名验证和若干次Hash运算)较好地解决了无线重编程中虚假广告DoS攻击和更新代码安全问题.

ADV-Data-Hash方案是基于广告、更新代码页的有限性和有序性的前提. 然而，无线重编程中的Req消息的发送是无法预测的，并且其中的包向量有2^{48}种组合，因此对不能用Hash散列链的方式解决，有待于进一步的研究.

参 考 文 献

1 Simon H, Ramkumar R, Roy SS, et al. Sensor network software update management: A survey. International Journal of Network Management, 2005, 15(4): 283—294

2 Wang Q, Zhu YY, Cheng L. Reprogramming wireless sensor networks: Challenges and approaches. In: IEEE Network, May 2006, 20(3): 48—55

3 张 羽, 蒋泽军, 周兴社. 无线传感器网络重编程技术研究. 计算机科学, 2008, 35(5): 66—68, 112

4 Hui J, Culler D. The dynamic behavior of a data dissemination protocol for network reprogramming at scale. In: Proc. of 2nd ACM SenSys' 04, Baltimore, Maryland, USA, 2004: 81—94

5 Lanigan PE, Gandhi R, Narasimhan P. Sluice: Secure dissemination of code updates in sensor networks. In: IEEE International Conference on Distributed Computing Systems, Lisbon, Portugal, 2006. 278—279

6 Dutta PK, Hui JW, Chu DC, et al. Securing the Deluge network programming system. In: ACM/IEEE Conference on Information Processing in Sensor Networks, Nashville, TN, April 2006: 326—333

7 Deng J, Han R, Mishra S. Secure code distribution in dynamically programmable wireless sensor networks. In: ACM/IEEE Conference on Information Processing in Sensor Networks (IPSN), Nashville, TN, April 2006. pp. 292—300

8 Zhang Y, Zhou XS, Ji YM, et al. Secure and DoS-Resistant network reprogramming in sensor networks based on CPK. 4th IEEE International Conference on Wireless Communications, Networking and Mobile Computing, 2008: 1—5

9 Perrig A, Szewczyk R, Tygar JD, et al. SPINS: Security protocols for sensor networks. Wireless Networks, 2002, 8(5): 521—534

10 Levis P, Lee N, Welsh M, et al. TOSSIM: Accurate and scalable simulation of entire TinyOS applications. In: ACM International Conference on Embedded Networked Sensor Systems, 2003. 126—137