一种非对等无线传感器网络环境中安全高效的混合密钥管理机制

王 刚 孙良旭 曾子维 杨 丹
(辽宁科技大学软件学院 鞍山 114051)

摘 要 密钥管理是实现WSN安全的首要功能。针对非对等无线传感网络中现有密钥管理存在的安全隐患和开销大等问题,提出一种安全高效的密钥管理机制。该机制包括一个基于椭圆曲线密码的轻量级签密算法,该签密不仅计算和通信代价小,还具有较好的前向安全性。基于提出的签密算法设计了完整的簇密钥管理机制,用于保证簇内通信的安全性。该机制基于簇基密钥构建各簇中的对称簇密钥,能够有效解决因簇首被俘而导致簇中所有感知节点失效的问题。为了更好地适应网络的动态性和扩展性,当网络节点变化和移动时,该机制能够基于簇密钥更新链有效地对簇密钥进行更新和维护。同时,还设计了簇密钥安全管理模型,模型能够根据网络环境安全威胁的不同,自适应进行簇密钥的更新,从而进一步提高了簇密钥更新的效率和算法性能。安全分析和性能对比表明,提出的机制在安全性和开销方面均胜于已有的其他密钥管理机制。

关键词 密钥管理,非对等WSN,椭圆曲线密码,签密,簇密钥
中图法分类号 TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.7.027

Secure and Efficient Hybrid Key Management Mechanism in Heterogeneous WSN

WANG Gang SUN Liang-xu ZENG Zi-wei YANG Dan
(Software School, University of Science and Technology Liaoning, Anshan 114051, China)

Abstract Key management is crucially important for all security goals in WSNs. For solving the security vulnerabilities and heavy overhead problems of the existing key managements in heterogeneous WSN, a key management mechanism was put forward. The mechanism includes an ECC-based lightweight sigcryption algorithm which can not only cost less computation and communication, but also have better forward security. An entire cluster key management protocol is designed based on the above sigcryption that can ensure the communication security in the cluster and use cluster base key to generate the cluster key in each cluster. The employment of cluster base key can effectively avoid all SNs being invalid when CH is captured. To adapt the dynamic and scalability characteristic of WSN, the cluster key can be effectively refreshed and maintained by utilizing the cluster key refresh chain. In addition, a cluster key security management model was proposed which can self-adaptively refresh the cluster key according to the change of network environment thread and further improve the cluster key refreshment performance. The contrast results show that the presented mechanism is better than other existing mechanisms in terms of security and protocol performance.

Keywords Key management, Heterogeneous WSN, Elliptical curve cryptography, Signcryption, Cluster key

1 引言

WSN能够在特殊的环境中使用传感器监视、响应和控制各种信息和事件。然而,传感器节点在能量、计算和网络带宽等资源方面显著受限。另外,WSN也具有较多其他特性,如节点具有移动性、需支持可扩展性、特殊的网络流量特征以及对多种类型的攻击具有不确定性反应。这些特征使得WSN中安全机制的设计非常重要且具有较大的挑战性[1-4]。密钥管理是实现网络安全性目标的首要功能,因为网络节点必须基于有效的密钥才能实现各种加解密和鉴别等安全算法。

传统观点认为,非对称密码机制会增加WSN网络负担。然而,最近ECC和IBC的发展使得公钥密码应用于WSN成为可能。这是由于160位密钥的ECC安全性与1024位的RSA相同,而且椭圆曲线密码已在MICA2或MICAz节点上实现,最近IBC也被嵌入在WSN中[5,6]。

目前,已有一些非对等无线传感器网络环境下的密钥管理机制。Riaz提出HHWSN安全密钥管理框架(SACK)[7]。SACK中每个SN与BS或CH之间都有对称密钥以建立点到点的安全通信,每个簇中所有SN共享一个特定的密钥用于簇内安全通信。然而,SACK中敌手能够使用主密钥以新

到稿日期:2015-06-04 返修日期:2015-08-10 本文受国家自然科学基金(61402213),辽宁科技大学校青年基金项目(2014QN19)资助。
王 刚(1978-),男,博士,副教授,主要研究方向为无线传感器网络、网络安全,E-mail:purgwg@163.com;孙良旭(1979-),男,博士生,主要研究方向为无线传感器网络、网络安全;曾子维(1963-),男,教授,主要研究方向为无线传感器网络、移动计算;杨 丹(1978-),女,博士,副教授,主要研究方向为数据挖掘、无线网络。

节点身份渗入网络，且敌手很容易窃听以明文方式传输的密钥生成初始化种子，从而计算出簇密钥。事实上，当一个 SN 被俘后，整个 WSN 网络安全都会受到影响。SACK 中的密钥更新算法所需的通信和计算开销也较大。

Du 提出基于路由驱动和 ECC 的密钥管理机制(RDEC)[8]。RDEC 中每个 SN 发送密钥请求消息到距离最近的 CH，然后由 CH 计算路径上相邻节点之间的共享密钥并基于原有路径发送给 SN。RDEC 基于 ECC 公钥密码机制实现，存在如下缺陷：1)由于 SN 是在网络初始化配置后再划分到各簇，因此每个 CH 都需保存网络中所有 SN 的公钥以用于后续的共享密钥的生成，而其所需的存储空间量对于 WSN 节点很难承受；2)SN 以明文形式发送密钥请求消息，所以敌手很容易篡改消息内容并欺骗 CH；3)每个 CH 在网络初始化配置过程中建立与 SN 相关联的密钥。因此，除了在此过程中配置好的 SN 外，新的 SN 无法再注册公钥并加入网络。

Mizanur[9]提出一个异构无线传感器网络中的密钥管理框架(PKAS)。PKAS 基于对变换和 IBCS 实现，其目的是使用基于身份的密码机制来提高 RDEC 性能。网络中每个节点(CH 或 SN)都在预配置阶段得到一个唯一的 ID 和两个不同的随机数。每个 CH 保存所有 SN 的 ID 和随机数，并以此来对簇内的 SN 进行认证。PKAS 机制中 SN 随机数由 BS 周期性更新并通过 CH 分发给 SN，这会给网络带来较大的通信负荷。而与之相比，本文提出的 SEKM 机制明显更为有效，这是因为 WSN 中网络传输的代价通常远大于计算所需的代价。另外，PKAS 中每个 SN 都需存储所有 CH 的 ID，而且只有建立簇以后才能进行认证，这样不仅增加了 SN 的存储负担，同时也降低了网络的扩展性和灵活性。

PIBK 也是一种非对等无线传感器网络中的基于身份的密钥管理协议。PIBK 针对静态网络设计，网络中所有 SN 均为静止且位置已知。SN 使用 IBC 建立对偶密钥。在预配置阶段，为每个 SN 分配网络密钥、簇密钥和 SN 密钥。之后，每个 SN 将在规定的引导时间内发送 ID 给它的邻节点。引导结束后，所有 SN 将保存邻居节点的 ID 以使得簇内任意两个节点之间都能够建立共享密钥。与其他机制相比，PIBK 机制中引导时间的设置十分关键。

Raazi 等提出 MUQAMI 机制[10]。虽然 MUQAMI 有密钥更新和节点撤销功能，但是它仅适用于位置已知的非对等无线传感器网络，即 BS 必须事先知道所有 SN 的地理位置。因此，MUQAMI 不支持节点移动，且由于 MUQAMI 中 SN 数量在预定义参数中被限制，这在一定程度上降低了协议的扩展性。

本文提出一个非对等网络环境下的混合密钥管理机制(Secure and Efficient Hybrid Key Management，SEKM)。非对等网络环境包含 3 类节点：Sink 节点、簇首节点和感知节点。3 类节点都有各自的作用，与感知节点相比，簇首节点性能更强，它通常承担更多的职责，如数据融合、簇组管理和任务调度等。设计了基于 ECC 的轻量级签密算法，该算法具有前向安全性和较小的存储和计算开销，被用于确保基站与簇首或各簇首之间的私密性和鉴别性。设计完整的簇密钥管理机制，用于保证簇内通信的安全性；该机制支持节点移动性，能够更好地适应 WSN 动态和扩展等特性。基于簇基密钥防止簇首被俘引起簇内所有感知节点失效，设计了簇密钥更新链和簇密钥安全管理模型，进一步减小了簇密钥更新所需代价，提高了协议的灵活性、扩展性及安全性。

2 SEKM 机制

2.1 预备知识

SEKM 采用混合密钥管理机制确保网络的安全性。网络中共包括 3 类节点：Sink 节点、簇首节点(Cluster Head，CH)和感知节点(Sensor Node，SN)。网络部署前由 Sink 在 CH 和 SN 中预置相关密钥。其中，Sink 和 CH 之间采用基于椭圆曲线密码的签密机制实现安全通信。网络部署后，CH 和 SN 之间采用基于簇密钥的轻量级对称密码机制确保簇内信息传输的安全性。各类节点中存储的密钥情况如表 1 所列。

表 1 各类节点中存储的密钥情况

Sink 节点		CH_i 节点		SN_j 节点	
U_{SK}	Sink 公钥	U_{SK}	Sink 公钥	K_{CH_i}	SN_j 簇密钥
V_{SK}	Sink 私钥	U_{CH_k}	CH_k 公钥	$K_{SN_j^i}$	SN_j 簇基密钥
U_{CH_i}	CH_i 公钥	U_{CH_i}	CH_i 公钥	K_{SN_j}	SN_j 对称密钥
V_{CH_i}	CH_i 私钥	V_{CH_i}	CH_i 私钥		
K_{SN_j}	SN_j 对称密钥	K_{CH_i}	CH_i 簇密钥		
		$K_{SN_j^i}$	SN_j 的簇基密钥		

SEKM 密钥管理框架所基于的网络模型假设包括：

(1)由于硬件限制，SN 不具备安全防护功能。

(2)与 SN 相比，CH 性能更好。CH 是 Sink 和 SN 之间的桥梁，具备非对称密码计算能力。

(3)每个 SN 和 CH 都有唯一的 ID。

(4)Sink 在计算、存储和电源供给等方面的能力不受任何限制，它具有更好的可靠性和安全性。

(5)CH 和 SN 节点可自由移动。

(6)网络中 CH 和 SN 节点通常配置在无人监管区域。SN 感知环境并发送原始数据给 CH，经数据融合等处理后，CH 将最终监测数据经多跳方式发送给 Sink 节点[11]。

2.2 轻量级签密算法

签密方法由加密和数据签名组成，可用于确保消息的私密性和可鉴别性。签密方法能够有效降低 WSN 中的计算、通信和存储开销。本文提出一个安全高效的轻量级签密算法，与其他签密比较[12]，本算法的计算和存储开销更低，并且具有前向安全性，即使 CH 的密钥泄露，也不影响之前传输消息的私密性。

在初始化网络配置时，Sink 基于 ECDLP 生成公私密钥对，然后将所需的对称和非对称密钥分别分配给 CH 和 SN。由于 CH 和 Sink 节点之间的通信链路的安全性更为重要，因此 SEKM 中 CH 和 Sink 之间基于提出的轻量级签密算法实现安全通信。表 1 中 V_{CH_i} 是 U_{CH_i} 以 G 为基的离散对数，网络中每个 CH 都有相邻 CH 的公钥 U_{CH_k}。Sink 使用 V_{SK} 和 U_{CH_i} 计算消息的签密，CH_i 接收到签密后基于 V_{CH_i} 和 U_{SK} 解密消息并验证其真实性。具体签密过程如下。

(1)签密

1)Sink 生成随机数 r_i；

2)计算 $R=r_i\times G=(r_1,r_2)$；

3)计算 $K=r_i\times U_{CH_i}=(k,l)$；

4)计算 $C=E_k(m)$；

5)计算 $s=V_{SK}-r_i \bmod q$；

6)发送的消息为：R,C,s。

(2)解密和验证

1)CH_i 节点接收到消息后，计算 $K=R\times V_{CH_i}=(k,l)$；

2)计算 $m=D_k(C)$；

3)如果 $s\times G-R=U_{SK}$，则 CH_i 接收明文 m，且 k 为会话密钥。在下一个周期内，CH_i 与 Sink 采用 k 为对称密钥进行通信。

2.3 簇密钥生成

为了避免簇首 CH_i 被俘泄露簇中感知节点 SN_j 的预置对称密钥 K_{SN_j}，仅有安全性较高的 Sink 节点存储 K_{SN_j}。为实现安全高效的簇组通信，在全网拓扑形成后，Sink 为 CH_i 分配簇基密钥，簇首以此来构建安全簇通信所需的簇密钥。具体步骤如下：

1)Sink 节点为 CH_i 簇中的节点 SN_j 生成簇基密钥 $K_{SN_j^i}$，并向簇首 CH_i 发送消息：$M=E_{K_{SN_j}}(K_{SN_j^i},CH_i,t.s.)\parallel Sgn(K_{SN_j^i},ID_{SN_j},t.s.)$，其中，$Sgn$ 为 2.2 节中的签密算法。

2)收到消息 M 后，CH_i 验证 M 的正确性。如果失败，丢弃 M；否则，生成簇密钥 K_{CH_i} 并使用解密出的 $K_{SN_j^i}$ 为 SN_j 生成消息：$M'=E_{K_{SN_j^i}}(K_{CH_i},CH_i,t.s.)\parallel(K_{SN_j^i},CH_i,t.s.)_{K_{SN_j}}$，然后将 M' 发送给 SN_j。

3)收到消息 M' 后，SN_j 使用密钥 K_{SN_j} 解密并认证簇基密钥 $K_{SN_j^i}$，之后基于 $K_{SN_j^i}$ 计算得到簇密钥 K_{CH_i}。

2.4 簇密钥更新

为了提高网络的可扩展性，在网络运行时应随时允许新节点的加入，同时也允许原节点退出网络或切换到其他簇。为保证簇密钥的前向安全性和后向安全性，在新节点加入和原节点失效时，应根据网络安全需求级别自适应地动态更新簇密钥。另外，SN 在簇间移动和切换时，应在新簇中注册并在原簇中注销，同时这也涉及了原簇和新簇簇密钥的更新。

(1)新 SN 加入

新节点 SN_n 根据簇生成算法选择加入簇 CH_i 时，需要向 CH_i 注册并由 CH_i 分配簇密钥。为保证后向安全性，即新加入节点不能知道原有的簇密钥，否则它就能利用其恢复簇内传输过的消息。为此，CH_i 簇中所有节点均需更新簇密钥。为减小簇密钥更新代价，在网络部署前由 Sink 节点为每个 SN 节点预置簇密钥更新链 F_c，用于辅助更新簇密钥。F_c 与传统的密钥池不同，它占用的节点存储空间小，且长度可以根据网络扩展容量自适应调节。

·簇密钥更新链管理

簇密钥更新链 F_c 用于实现轻量级簇密钥更新。一方面，F_c 可以提高簇密钥更新效率，降低簇密钥更新所需要的计算和通信开销；另一方面，F_c 仅会占用 SN 节点较少的存储空间。为此，设计了具有可伸缩性的合理的 F_c 管理方式。

设网络中节点数为 n，簇数为 m，则每个簇中感知节点的平均数为 $\bar{k}=n/m$。如果网络扩展系数为 x，则每个簇中加入节点或退出节点的平均数应为 $\bar{k}'=n/m*x(0\leqslant x<1)$。因此，确定 F_c 的长度 L 为：$L=n/m\times x+a$。a 为密钥冗余长度。考虑到节点的移动也会引起节点加入或退出，设节点跨簇移动比例为 c，修正 F_c 的长度为：$L=n/m\times(x+c)+a$，其中参数 x，c 和 l 的具体数值可以根据不同应用场景灵活设置。为避免 F_c 长度限制簇密钥更新，从而影响网络安全性或可扩展性，当安全等级和潜在网络危险较低时，F_c 中的簇密钥更新因子可重复使用；反之，可以由 Sink 节点在线部署新的簇密钥更新链 F_c'。F_c' 的长度 $L'=r\times L$，r 为扩展延长系数。在线部署 F_c' 时，使用基于簇密钥的对称密码体制建立安全通道，确保 F_c' 部署的安全性。

·节点加入

新节点 SN_n 加入簇 CH_i 时需要先注册，以实现簇首和新节点之间的双向认证并生成簇基密钥 $K_{SN_n^i}$。同时，为保证前向安全性，簇中所有节点均需更新簇密钥。具体步骤如下：

1)CH_i 簇中任意节点 SN_j 依次选用 F_c 中簇密钥更新因子 f_k 计算新簇密钥 $K'_{CH_i}=H_{f_k}(K_{CH_i})$。其中，$H$ 为单向哈希函数。

2)SN_n 向簇首 CH_i 发送注册申请消息：$M=\{H_{K_{SN_n}}(SN_n,t.s.),SN_n,t.s.\}$。

3)CH_i 收到 M 后，计算消息 $M'=E_{H(k)}(M,SN_n,t.s.)$ 并发送给 Sink 节点，其中，k 为执行签密算法过程中 Sink 发送给 CH_i 的会话密钥。

4)收到消息 M' 后，Sink 首先解密 M' 并验证 M' 和 M 的正确性，如验证失败，则丢弃消息 M'，表明 SN_n 为非合法节点；否则，按照 2.3 节中的方式，计算并发送 SN_n 的簇基密钥 $K_{SN_n^i}$ 给 CH_i。

5)CH_i 发送 M' 给节点 SN_n，$M'=E_{K_{SN_n^i}}(K'_{CH_i}\parallel meta)\parallel E_{K_{SN_n}}(K_{SN_n^i},CH_i,t.s.)$，其中 $meta$ 为固定值。

6)SN_n 验证消息的正确性后，成功计算并得到簇基密钥 $K_{SN_n^i}$ 和新簇密钥 K'_{CH_i}。

(2)原 SN 退出

为保证前向安全性，确保节点退出之后不能再解密获得簇中传输的消息，簇密钥必须及时更新。为了提高更新效率，采用扰动因子策略进行簇密钥的更新。其思想是：簇首 CH_i 随机生成扰动因子并使用簇成员的簇基密钥进行加密，然后发送给每个簇成员。具体过程为：

1)簇首 CH_i 随机生成扰动因子 d，并使用簇成员的簇基密钥 $K_{SN_n^i}$ 加密，生成消息 $M=E_{K_{SN_j^i}}(CH_i,d)$；然后，将 M 发送给节点 SN_j。

2)SN_j 收到消息 M 后，解密并验证消息 M，验证正确后接受扰动因子 d，并基于 d 选取簇密钥更新因子 f_d。

3)计算得到 f_d 后，SN_j 重新计算新簇密钥：$K'_{CL_i}=H_{f_d}(K_{CL_i})$。

(3)SN 移动

由于网络中节点具有移动性，因此 SN 可能会移动到其他簇。为了提高路由能效，应重新选择与新簇首 CH_k 关联的路径作为移动节点 SN_m 到 Sink 的最短路径。同时，SN_m 需向新簇首 CH_k 注册并从原簇注销。具体步骤如下：

1)移动节点 SN_m 注册新簇，即向新簇首 CH_k 发送消息 $M=M_1\parallel M_2$。其中，$M_1=E_{SN_m^i}(SN_m,CH_k,t.s.)$，$M_2=(SN_m,CH_i,t.s.)$。

2)收到消息 M 后，CH_k 计算 $M'=Sgn(SN_m,CH_k,t.s.')\parallel M_1$，并将 M' 发送给 SN_m 的原簇首 CH_i。

3)CH_i 验证消息 M' 的正确性后，向 CH_k 发送消息 $M''=Sgn(K_{SN_m^i},SN_mCH_i,t.s.'')$；否则，发送错误报警消息；同时，

将节点 SN_m 从簇中删除。

4)如收到错误报警消息,则 CH_k 将 SN_m 列入黑名单;否则,解密并验证消息 M'',然后发送 $E_{SN_m^i}(K'_{CH_k}, SN_m, CH_k, t.s.)$ 给 SN_m。

5)SN_m 首先验证消息的正确性,然后解密得到新簇的簇密钥 K'_{CH_k},且簇 CL_k 中所有节点按照 2.4 节中的方式更新簇密钥 K'_{CH_k};

2.5 簇密钥安全管理模型

WSN 中很多应用和工作场景都存在不同程度的安全隐患,采用一定的安全机制可以有效地抵御网络威胁和攻击,保证数据传输的安全性。然而,安全机制的引入必然会增加网络系统负荷,消耗一定的网络和系统资源。为了最大程度地确保网络的安全性,尽可能减少安全机制所消耗的计算和通信资源,设计了簇密钥安全管理模型(KSMM)。KSMM 综合考虑网络信息敏感度、网络安全强度和节点信任度等安全因素,有效地对资源消耗较多的簇密钥更新过程进行优化,目的是保证网络基本安全需求的同时最大化网络生存寿命。

定义 1(网络信息敏感度 NIS) 任意网络任务信息按照内容敏感度可分为 3 类:1)NIS_A:敏感信息;2)NIS_B:信息中含一定敏感因素;3)NIS_C:不敏感。

定义 2(网络安全强度 NSS) 根据网络应用范围和场景,NSS 共分为 5 个级别($NSS_0 - NSS_4$)。其中,数值越小,要求的安全强度越高。

定义 3(节点信任度) 每个节点采用文献[13]的方式建立节点信任度,节点信任度值范围为[0,1],数值越大,说明节点越可信。

簇密钥的更新会消耗一定的节点计算和网络带宽资源,特别地,当节点移动较为频繁时,簇密钥更新代价会显著增加。然而,当某些网络安全强度需求较低,传输数据敏感度较低,或是变化节点信任度较高时,频繁地更新簇密钥只会带来不必要的资源损耗,并不会改变网络的安全性。为此,KSMM 综合多种网络安全因素,仅在必要(即网络安全性受到一定威胁)时进行簇密钥更新。设簇 CH_i 有节点变化,此时,CH_i 正在执行的任务(如监控、定位等)中涉及到的信息的敏感度量化值为 x_{NIS},其工作场景对应的网络安全强度级别量化值为 x_{NSS},发生变化的节点信任度为 x_{TR},则簇首 CH_i 根据 $d_{KF_i} = x_{NIS} * \alpha + x_{NSS} * \beta + x_{TR} * \gamma$ 计算簇密钥更新度,其中,α,β,λ 为权值系数,$\alpha+\beta+\lambda=1$。此时,如果 $d_{KF_i} \geqslant \bar{d}_{KF}$($\bar{d}_{KF}$ 为簇密钥更新阈值),则说明虽然簇内节点发生变化,但节点变化并不会危及簇内通信安全,因此不需要更新簇密钥;否则,按照上述方式对簇密钥进行更新。

3 安全分析与性能比较

3.1 安全性分析

定理 1 如果 ECDLP 是计算上不可行的,则 SEKM 签密机制是安全的。

证明:由 2.2 节已知,密钥 k 是点 K 坐标系上的 x 值。敌手需要正确计算下式才能得到 k 值并破坏 SEKM 的安全性:$K = r_i \times U_{CH_i} = (k,l)$ 或 $K = R \times V_{CH_i} = (k,l)$。

敌手虽然知道 U_{CH_i},G 和 R,但是必须先解决 ECDLP 难题才能得到 k。证毕。

定理 2 SEKM 签密机制中 CH 是难以追踪的,即使 CH 私钥被破解,敌手也不能解密 CH 发送过的签密消息。

证明:前向私密性意味着敌手即使得到 CH 唯一的私钥,也不能破解传输的明文。即便敌手得到 V_{CH_i},它也不能重新计算出密钥 k 并解密密文 C。这是因为,虽然敌手知道 C,$R=(r_1,r_2)$,s' 和 V_{CH_i},但是由于 r_i 是每次会话使用的不同的未知随机数,而它不可能基于公式 $s = P_{tk} - r_i \bmod q$ 生成 r_i,因此敌手不能解密 CH 发送过的签密消息。证毕。

定理 3 SEKM 中 SN 不会因为 CH 被俘而失效。

证明:在预置密钥阶段,Sink 为每个 SN 初始化唯一的对称密钥。在 SACK 等机制中,CH 存储所有簇内 SN 节点的对称密钥,一旦 CH 被俘,则簇内所有 SN 节点会因为对称密钥的泄露而无法再建立安全信道,在信息较为敏感的网络环境中这会导致 SN 直接失效。然而,在 SEKM 中 CH 使用簇基密钥构建安全群组通信环境,能够唯一鉴别 SN 安全身份的对称密钥仅存储在安全级别高的 Sink 节点中,由于 CH 无法直接推导 $K_{SN_j^i} \rightarrow K_{SN_j}$,因此 SN 不会因为 CH 被俘而泄露 K_{SN_j} 和失效。证毕。

定理 4 SEKM 机制能够有效抵御重放、篡改和伪造等攻击。

证明:由于 Sink 和 CH 分别基于对称密钥和簇基密钥对 SN 身份进行验证和加解密,因此敌手无法证明自己是一个可信的 SN,且 SEKM 中利用时间戳可以验证 $t.s.' - t.s. \leqslant \Delta T$ 是否成立来防止重放和拒绝服务攻击。Sink-CH 之间的签密机制、CH-SN 之间基于簇基密钥的安全通信,以及基于 K_{CH_i} 的安全簇内通信,可以确保所有节点都能彼此认证,因此 SEKM 机制能够有效地抵御篡改和伪造攻击。证毕。

3.2 性能分析与比较

从预置密钥数和计算量等方面,将 SEKM 与诸多经典机制进行比较。假设网络中 SN 和 CH 的数量分别为 N 和 M,SN 随机地分布在各簇中,且基于 SEKM 初始化过程对 SN、CH 和 Sink 预置对应密钥。

表 2 显示了网络中包含 $N+M$ 个节点时,SEKM 在存储空间需求方面与其他机制的对比情况。假设 CH 的数量为 20,SN 的数量在 1~250 之间变化,基于表 2,图 1 给出了几种机制对存储空间的需求比较结果。如图 1 所示,与其他机制相比,SEKM 机制需要的存储空间最少。

表 2 存储空间需求

机制	密钥所需存储空间
SEKM	6M+3N
SACK	5M+5N
RDEC	M(N+3)+2N
PKAS	4M+6N
PIBK	4M+4N

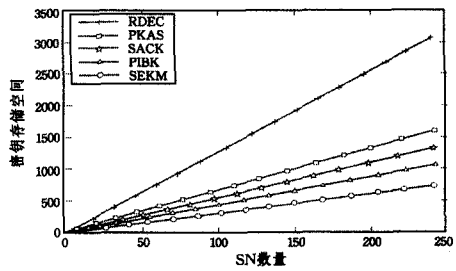

图 1 存储空间对比

(下转第 185 页)

Springer Berlin Heidelberg,2008:58-71

[6] Xiang J,Yanoo K,Maeno Y,et al. Automatic synthesis of static fault trees from system models[C]// 2011 Fifth International Conference on Secure Software Integration and Reliability Improvement (SSIRI). IEEE,2011:127-136

[7] Lauer C,German R,Pollmer J. Fault tree synthesis from UML models for reliability analysis at early design stages[J]. Acm Sigsoft Software Engineering Notes,2011,36(1):1-8

[8] Hu W,Deng Z,Hong Y. A method of FTA base on UML use case diagram[C]// 2011 9th International Conference on IEEE Reliability,Maintainability and Safety (ICRMS). 2011:757-759

[9] Harper D C. Fault Tree Analysis of UML Designs[J]. Technometrics,2012,19(3):346-347

[10] Tiwari S,Gupta A. An Approach to Generate Safety Validation Test Cases from UML Activity Diagram[C]// 2013 20th Asia-Pacific Software Engineering Conference. IEEE,2013:189-198

[11] Zhao Z. UML Model to Fault Tree Model Transformation for Dependability Analysis[D]. Carleton University Ottawa,2014

[12] Li S, Li X. Study on generation of fault trees from Altarica models[J]. Procedia Engineering,2014,80:140-152

(上接第 156 页)

除存储空间外,SEKM 机制还具有其他显著的性能优势,如计算和通信负载少、支持 SN 移动、抵御 CH 被俘、基于签密确保 Sink-CH 之间的安全通信等。将 SEKM 机制与 SACK、RDEC、PKAS 和 PIBK 机制进行了比较,如表 3 所列。PKAS 与 SEKM 机制具有最少的缺点,RDEC、PKAS 和 PIBK 机制中,SN-CH 之间的链路使用 PKC 机制,这种方式增加了 SN 的计算量,且 SEKM 机制和 RDEC 机制不需要事先分簇。

表 3 5 种机制性能的对比

特点	机制				
	SACK	RDEC	PKAS	PIBK	SEKM
节点移动性	无	无	无	无	有
SN 中密钥数	3	2	3	4	3
单个 SN 被俘	网络失效	多个 SN 失效	仅单个 SN 失效	整个网络失效	仅单个 SN 失效
SN 与 CL 差别	无	有	有	无	有
PKC 使用形式	加密	加密和签名	加密	密钥协商	签密
PKC 使用范围	CH-Sink	SN-CH CH-Sink	SN-CH CH-Sink	SN-CH CH-Sink	CH-Sink
可扩展性	支持	支持	支持	支持	支持
密钥管理基于分簇	是	否	是	是	否
簇密钥更新效率	低	无	无	低	高

结束语 密钥管理是 WSN 安全的基础。近年来,针对非对等无线传感器网络已设计了一些密钥管理机制。本文提出一种新颖、高效和安全的混合密钥管理机制 SEKM。SEKM 机制在安全性和性能方面具有显著优势和特征,包括在 CH 和 Sink 之间使用基于椭圆曲线密码的签密机制确保数据传输的安全性,提出的签密机制具有前向安全性和公开验证特点,并且协议开销小。针对 SN 的加入、退出和移动,提出轻量级的簇密钥生成和更新算法,不仅资源和能量消耗少,而且基于簇基密钥方式可以有效避免因簇首被俘而导致簇中所有 SN 节点甚至整个网络失效。实际上,SEKM 不仅具备较好的安全属性,而且计算、通信和存储开销较小。

参 考 文 献

[1] Zeng Wei-ni, Lin Ya-ping, Yu Jian-ping, et al. Group Key Management Based on Random Perturbation in Wireless Sensor Networks[J]. Journal of Software,2013,24(4):873-886(in Chinese)
曾玮妮,林亚平,余建平,等. 传感器网络中基于随机混淆的组密钥管理机制[J]. 软件学报,2013,24(4):873-886

[2] Guo Song-hui, Niu Xiao-peng, Wang Yu-long. Elliptic Curve Based Light-weight Authentication and Key Agreement Scheme [J]. Computer Science,2015,42(1):137-141(in Chinese)
郭松辉,牛小鹏,王玉龙. 一种基于椭圆曲线的轻量级身份认证及密钥协商方案[J]. 计算机科学,2015,42(1):137-141

[3] Wang Gang,Wen Tao,Guo Quan,et al. An Efficient and Secure Group Key Management Scheme in Mobile Ad Hoc Networks [J]. Journal of Computer Research and Development,2010,47(5):911-920(in Chinese)
王刚,温涛,郭权,等. 移动自组网中安全高效的组密钥管理方案[J]. 计算机研究与发展,2010,47(5):911-920

[4] Lee J,Kapitanova K,Son S H. The price of security in wireless sensor networks [J]. Computing Network Journal, Elsevier, 2010,54(17):2967-2978

[5] Oliveira L B, Aranha D F, Gouvea C P L, et al. TinyPBC: Pairings for authenticated identity-based non-interactive key distribution in sensor networks [J]. Computing Communication Journal,Elsevier,2011,34(3):485-493

[6] Hagras E A,El-Saied D,Aly H H. Energy efficient key management scheme based on elliptic curve signcryption for wireless sensor networks [C]// 28th National Radio Science Conf(NRSC). 2011

[7] Riaz R, Naureen A, Akram A, et al. A unified security framework with three key management schemes for wireless sensor networks[J]. Computing Communication Journal,Elsevier,2008(31):4269-4280

[8] Du X,Guizani M,Xiao Y,et al. A routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks [J]. IEEE Transactions Wireless Communication,2009,8(3):1223-1229

[9] Mizanur R,Sk M,El-Khatib K. Private key agreement and secure communication for heterogeneous sensor networks [J]. Parallel Distribution Computing,Elsevier,2010(70):858-870

[10] Khaliq-ur-Rahman Raazi S M,Lee H,Lee S,et al. MUQAMI+: A scalable and locally distributed key management scheme for clustered sensor networks [J]. Annalso of Telecommunications, Springer,2010,65(1/2):101-116

[11] Stavrou E,Pitsillides A. A survey on secure multipath routing protocols in WSNs [J]. Computing Network Journal,2010(54):2215-2238

[12] Alagheband M R,Aref M R. Dynamic and secure key management model for hierarchical heterogeneous sensor networks [J]. IET Information Security,2012,6(4):271-280

[13] Zhan Guo-xing,Shi Wei-song,Deng Ju-lia. Design and implementation of TARFA trust-aware routing framework for WSNs [J]. IEEE Transactions on Dependable and Secure Computing, 2012,9(2):184-197