第38卷 第24期
Vol.38 No.24

计 算 机 工 程
Computer Engineering

2012年12月
December 2012

·安全技术·

文章编号：1000—3428(2012)24—0115—04 文献标识码：A 中图分类号：TP393

无线传感器网络途中过滤增强方案

任秀丽，张 晨

(辽宁大学信息学院，沈阳 110036)

摘 要：在无线传感器网络中，节点被俘获后会向网络中注入大量虚假数据。为此，提出一种途中过滤增强方案。使用加密密钥和验证密钥防止途中节点篡改数据，采用安全性增强方案解决途中节点遭到破坏而无法传递和检测数据的问题，利用备份节点的密钥验证转发数据的正确性，由此过滤虚假数据，并引入 *MAX_FALSE* 参数，消除不完全虚假数据对基站接收数据的影响。仿真结果表明，与 SEF、DEF、FIMA 相比，该方案的过滤能力更强，能耗更少。

关键词：无线传感器网络；途中过滤；虚假数据；过滤增强

En-route Filtering Enhancement Scheme in Wireless Sensor Network

REN Xiu-li, ZHANG Chen

(College of Information, Liaoning University, Shenyang 110036, China)

【Abstract】 In Wireless Sensor Network(WSN), nodes will inject large amount of false data into the network if they are captured. This paper proposes an en-route filtering enhancement scheme to solve this problem. It employs encryption keys and authentication keys to prevent en-route nodes from distorting data. When en-route nodes are destroyed and can not transfer or detect data, a safety enhanced scheme is applied, which uses the encryption keys of backup nodes to validate the authenticity of transferred data and discard false ones. Parameter *MAX_FALSE* is introduced to eliminate the influence that incomplete false data make on the received data of base stations. Simulation results show that the scheme is more effective and energy efficient compared with SEF, DEF and FIMA.

【Key words】 Wireless Sensor Network(WSN); en-route filtering; false data; filtering enhancement

DOI: 10.3969/j.issn.1000-3428.2012.24.028

1 概述

无线传感器网络(Wireless Sensor Network, WSN)经常部署在不安全的环境中，传感器节点时刻受到攻击、捕获等威胁[1]。攻击者利用被俘获的节点向网络中注入大量的虚假数据，欺骗数据收集中心做出错误的决策，耗尽节点的能量，因此，网络的安全性面临着巨大考验[2]。为了能够及时检测、丢弃虚假数据，保证网络的正常运行，本文提出了无线传感器网络途中过滤增强方案 EEF(Enhanced En-route scheme for Filtering false data injection)。

2 相关工作

为了应对虚假数据注入攻击，文献[3]通过拓扑控制，完成对检测区域的覆盖，但每个节点存储的密钥数量很大，抗俘获性较差。SEF 方法过滤能力有限，而且只允许少量的节点被俘获[4]。DEF 算法利用 Hill Climbing 方式，将簇内感知节点的验证密钥传递给途中节点[5]。但是 DEF 方案需要根据拓扑的改变定期进行重新广播，耗费能量。FIMA[6]在 IHA[7](Interleaved Hop-by-hop Authentication)方案的基础上使用 fuzzy logic system 将途中节点分为转发节点、验证节点和跳跃节点，因此，虚假数据通过的跳数将会增加。文献[8]将途中过滤与溯源追踪相结合，过滤效率不高。文献[9]方案中节点在 2 种模式间切换，能量消耗较高。

在本文提出的 EEF 方案中，感知节点生成验证和加密 2 种密钥。节点使用加密密钥加密数据，途中节点使用验证密钥验证数据的真实性；当途中节点遭到破坏时，使用安全性增强方案，利用备份节点中的密钥验证转发数据的正确性，过滤掉虚假数据，减少能量的消耗。

3 途中过滤增强方案

3.1 相关定义

本文方案中涉及的定义如下：

定义 1 不完全虚假数据是指簇内被俘获的感知节点在生成数据发给簇头节点时，数据中加密的事件是真实的，但是其使用的验证密钥是篡改的验证密钥，从而导致途中节点丢弃的数据。基站节点 *BS* 无法获知簇内发生的事件。

定义 2 安全临界值 *MAX_FALSH* 是根据网络安全要

基金项目：辽宁省自然科学基金资助项目(201202089)；辽宁大学“211 工程”三期建设基金资助项目“极端计算技术项目”

作者简介：任秀丽(1965－)，女，教授、博士后，主研方向：无线网络与通信，网络安全；张 晨，硕士研究生

收稿日期：2012-03-15 **修回日期：**2012-04-23 **E-mail：**zc-zhangchen522@163.com

求事先定义的参数。只有当发现的虚假验证密钥数达到 *MAX_FALSH* 值时，才丢弃数据包。

定义 3 虚假验证密钥数 *FALSE* 是封装于数据包中、初始值为 0 的参数。

3.2 方案的实现

网络中的节点部署完成后，簇头节点收集簇内每个感知节点的验证密钥，发送给由簇头节点到 *BS* 路径上的所有途中节点。途中节点在接收、存储验证密钥后，实行安全性增强方案。簇内节点用加密密钥加密要传递的数据。途中节点接收数据，根据之前收到的验证密钥验证数据，丢弃虚假数据。由于途中节点不知道感知节点的加密密钥，无法篡改数据，因此保证了数据的安全性。

本文方案分为密钥生成、验证密钥分发、数据生成、途中过滤和 *BS* 检测 5 个阶段。

3.2.1 密钥生成

节点在分发之前，内部只存储了 1 个种子 k、3 个 hash 函数 p、g 和 w、随机分配的安全密钥 h 和公共密钥 c。以簇内节点 n_2 为例，如图 1 所示。$k_{n_2}^m$ 表示节点 n_2 中的种子；$h_{n_2}^o$ 表示随机分配的安全密钥；c 表示公共密钥；*CH* 表示簇头节点；n_i 表示簇内感知节点(i=1, 2,…, l)；EN_i 表示途中过滤节点(i=1, 2, …, q)。加密密钥 $k_{n_2}^1$ 是由种子 $k_{n_2}^m$ 通过 hash 函数 p 第 $m-1$ 次运算作用得到的，$k_{n_2}^1 = p^{m-1}(k_{n_2}^m)$。验证密钥 $z_{n_2}^1$ 是由种子 $k_{n_2}^m$ 通过 hash 函数 g 第 $m-1$ 次运算作用得到的，$z_{n_2}^1 = g^{m-1}(k_{n_2}^m)$。使用时，首先使用 $k_{n_2}^1$ 和 $z_{n_2}^1$。加密密钥和验证密钥生成之后，节点删除 2 个 hash 函数和种子 k。

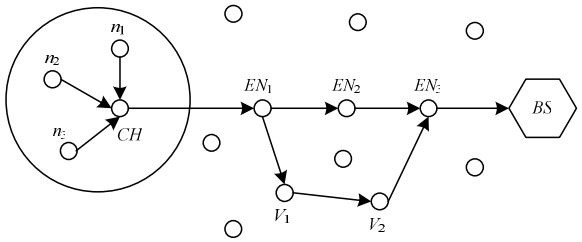

图 1 途中过滤过程

3.2.2 验证密钥分发

每个簇内感知节点将生成的验证密钥用安全密钥加密之后发给簇头节点。感知节点发送的信息 M 如下所示：

$$M=\{n_j,\ i_j,\ u,\ MAC(h_{n_j}^u, z_{n_j}^{i_j})\} \tag{1}$$

其中，n_j 代表节点 ID；i_j 表示节点 n_j 使用 hash 函数 g 产生的第 i_j 个验证密钥；$h_{n_j}^u$ 是安全密钥；u 表示 $h_{n_j}^u$ 在安全密钥链上的第 u 个密钥；$MAC(h_{n_j}^u, z_{n_j}^{i_j})$表示使用安全密钥 $h_{n_j}^u$ 对验证密钥 $z_{n_j}^{i_j}$ 进行加密后的数据。

簇头节点收集整个簇内节点发送的信息，整合起来产生密钥数据 $K(n)$：

$$K(n)=\{n_1,\ i_1,\ s,\ MAC(h_{n_1}^s, z_{n_1}^{i_1}),\ n_2,\ i_2,\ r,\ MAC(h_{n_2}^r, z_{n_2}^{i_2}),\cdots,$$
$$n_j,\ i_j,\ u,\ MAC(h_{n_j}^u, z_{n_j}^{i_j})\} \tag{2}$$

其中，n 表示簇 n。

密钥信息 $K(n)$生成之后，簇头节点向途中节点发送 $K(n)$。如图 1 中的 EN_1、EN_2、EN_3 为途中节点。以途中节点 EN_2 为例，EN_2 接收到 $K(n)$后首先查看自己的安全密钥 $h_{EN_2}^v$，依次与 $K(n)$中的安全密钥进行比较。例如，将节点 EN_2 安全密钥位于安全密钥链上的位置 v 与 $K(n)$中的 u 进行比较，若 $v<u$，则节点 EN_2 利用 hash 函数 w 和安全密钥 $h_{EN_2}^v$ 计算 $h_{n_j}^u$：$h_{n_j}^u = w^{(u-v)/10}(h_{EN_2}^v)$。

此处 hash 函数 w 产生的安全密钥每 10 个进行一次循环，因此，$h_{n_j}^u$ 的计算不会过于复杂。

用 $h_{n_j}^u$ 解密 $MAC(h_{n_j}^u, z_{n_j}^{i_j})$，最终得到验证密钥 $z_{n_j}^{i_j}$。将 $z_{n_j}^{i_j}$ 和 $z_{n_j}^{i_j}$ 对应的 n_j、i_j 存储下来。同时，将 u、$MAC(h_{n_j}^u, z_{n_j}^{i_j})$ 用 v、$MAC(h_{EN_2}^v, z_{n_j}^{i_j})$替换。如此，后面的途中节点越来越难以解密验证密钥，保证越接近簇头的途中节点存储的验证密钥越多，节点很快过滤掉虚假数据。并且防止越接近 *BS* 的途中节点因存储大量验证密钥而导致数据溢出。

一旦途中节点受损，节点存储的验证密钥也将消失，失去了过滤能力。其他途中节点继续传递虚假数据，浪费了能量。针对途中节点受损的情况，本文提出了安全性增强方案。以图 1 的 EN_2 节点为例，安全性增强方案实施过程如下：

(1)EN_2 将能解密的验证密钥 z 用公共密钥 c 加密，将加密后的数据 $MAC(c, z)$和其下一节点 EN_3 的 ID 发送给前一节点 EN_1。

(2)EN_1 接收数据，在其邻居节点列表$\{V_1, V_2,\cdots, V_n\}$中利用右手法则[10]找到最近的邻居节点 V_1。

(3)EN_1 存储 V_1 节点的 ID，并将接收到的 $MAC(c, z)$、EN_3 的 ID 发送给节点 V_1。

(4)EN_2 继续向下传递密钥数据 $K(n)$。

此方案保证了如果节点 EN_2 受损，备份节点 V_1 可以使用 EN_2 中的验证密钥验证数据。

3.2.3 数据生成

簇内感知节点感知到事件 E 的发生后将事件发给簇头。以簇内感知节点 n_i 第 1 次感知到事件为例，n_i 使用加密密钥 $k_{n_i}^1$ 和验证密钥 $z_{n_i}^2$ 对事件加密。将加密数据 M^{n_i} 传给簇头节点。

$$M^{n_i}=\{E,\ n_j,\ MAC(k_{n_i}^1,\ E), z_{n_i}^2,\ 2,\ FALSE(n_i)\} \tag{3}$$

其中，E 表示事件；2 表示使用的验证密钥链上的第 2 个密钥；初始 $FALSE(n_i)$值为 0。

簇头节点收集到 t 个簇内节点发送的数据，组成数据包 $M(n)$，发送给途中节点。

$$M(n)=\{E, n_{i_1}, MAC(k_{n_{i_1}}^1, E),\ z_{n_{i_1}}^2, 2, FALSE(n_{i_1}), n_{i_2},$$
$$MAC(k_{n_{i_2}}^1, E),\ z_{n_{i_2}}^2, 2, FALSE(n_{i_2}),\cdots, n_{i_t},$$
$$MAC(k_{n_{i_t}}^1, E),\ z_{n_{i_t}}^2, 2, FALSE(n_{i_t})\} \tag{4}$$

其中，n 表示簇 n；$n_{i_1}, n_{i_2},\cdots, n_{i_t}$ 代表发送数据的 t 个节点。

3.2.4 途中过滤

途中节点接收到数据包 $M(n)$，为了确保数据的正确

性，需要执行以下过程过滤掉虚假数据：

(1)查看 $M(n)$中是否包括 t 个不同的节点，如果包括，则保存；否则，丢弃。

(2)查看 $M(n)$中是否包括 t 个不同的验证密钥 z，如果包括，则保存；否则，丢弃。

(3)查看 $M(n)$中是否包括 t 个不同的加密数据 MAC，如果包括，则保存；否则，丢弃。

(4)查看自身在验证密钥分发阶段是否存储了 $M(n)$中某几个节点的验证密钥，如果有，比较是否相等。

以节点 n_{i_t} 为例，若途中节点在密钥分发阶段存储了 n_{i_t} 的验证密钥 $z^1_{n_{i_t}}$，比较 $z^1_{n_{i_t}}=g^{(2-1)}(z^2_{n_{i_t}})$ 是否成立：如果不成立，对应的 $FALSE$ 值加 1，再计算 $M(n)$中各个 $FALSE$ 的和是否小于等于预设参数 MAX_FALSH，如果小于等于，则保存；否则，丢弃。若成立，用 $z^2_{n_{i_t}}$ 覆盖途中节点存储的 $z^1_{n_{i_t}}$，继续向下传递 $M(n)$。

途中节点验证数据包是正确的继续向下传递。节点使用广播通信，周围邻居都能监听到节点是否向下传送数据包。若节点监听不到，则其下一节点受损，无法向下传递。此时途中节点启动安全增强方案。以图 1 中的节点 EN_1 为例，安全增强方案具体实现过程如下：

(1)EN_1 查看信息，显示 EN_2 的下一节点是 EN_3。

(2)EN_1 向 V_1 发送修复信息，V_1 查看自身邻居列表里是否有 EN_3，如果有，链路修复完毕，由 EN_1-V_1-EN_3 代替原来的 EN_1-EN_2-EN_3；如果没有，V_1 继续用右手法则寻找，直到找到一个节点，其邻居列表中有 EN_3，此时链路修复完毕，由 EN_1-V_1-V_2-EN_3 代替原来的 EN_1-EN_2-EN_3。

(3)EN_1 将改变后的链路信息发送给其前一节点即簇头节点 CH。

(4)链路修复完毕，作为 EN_2 的备份节点，V_1 充当 EN_2 的角色，执行对数据包的过滤功能。

3.2.5　*BS* 过滤

数据最后传递到 *BS* 节点，因为 *BS* 节点在部署之前存储了所有节点的密钥，能对数据中所有节点的加密数据进行最终验证。*BS* 节点使用相应的加密密钥解密，查看事件 E 是否在误差范围内。验证每个节点的验证密钥，验证步骤同途中过滤阶段。

4　仿真和性能分析

为了评价 EEF 方案的性能，将其与 SEF、DEF、FIMA 在过滤能力和能量消耗 2 个方面进行了比较。实验环境配置为 100 个节点均匀分布在 100 m×100 m 的区域里，节点初始能量为 1 J。数据包需要 5 个节点认证才能向下传送，相关参数 t 设置为 5，被俘获的节点数 N 设置为 4。

图 2 给出了在 N=4 时 4 种方案对虚假数据的过滤能力比较。从中可以看到，随着虚假数据通过的节点数的增加，对虚假数据的过滤能力不断增强。在 FIMA 方案[4]中，由于其参数 Tf 表示伪造的 MAC 阈值，Tf=1 时的过滤能力好于 Tf=2，因此取 Tf=1 进行实验比较。由于 EEF 在验证密钥分发阶段，途中节点根据随机分配的安全密钥获取验证密钥，因此越接近簇头的节点获得的验证密钥越多，验证数据包的能力越强。所以，在参数 MAX_FALSH 设置为 0 的情况下，EEF 方案在 10 跳以内过滤概率基本上达到 90%，高于 DEF、FIMA 和 SEF 方案。MAX_FALSH 设置为 2 时，因为允许途中节点继续传递检验出的虚假数据，所以 EEF 的过滤能力低于 DEF 方案，但仍高于 SEF 和 FIMA 方案。

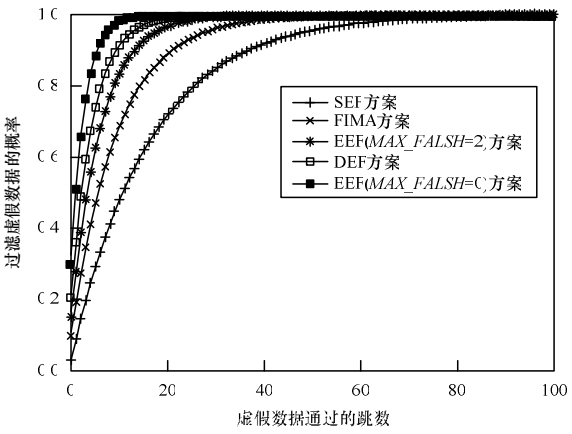

图 2　虚假数据的途中过滤能力比较

图 3 反映了在途中节点受损时虚假数据进入 *BS* 的情况。随着节点受损个数的增加，进入 *BS* 的虚假数据呈上升趋势。EEF 方案中参数 MAX_FALSH 设置为 0。在各方案中，SEF 的虚假数据通过率最高，途中节点受损，失去了过滤能力，无法检测丢弃虚假数据。其他方案对于节点的受损采取了一些措施，但本文采用的安全性增强方案不仅能够保证数据的传送能力，而且过滤能力不因途中节点的受损而降低，可以防止虚假数据进入 *BS* 节点。

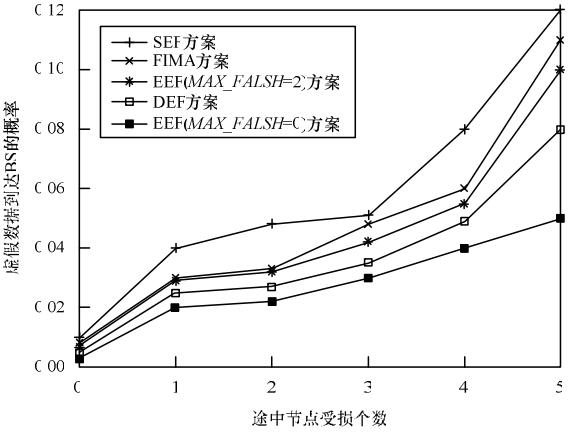

图 3　途中节点受损时虚假数据进入 *BS* 的概率比较

图 4 给出了在 N=4 的情况下 4 种方案的能量消耗情况。虚假数据包被途中节点转发时造成能量消耗。途中节点过滤能力弱和被俘获节点的增多都会使网络的能耗变大。由图 4 可知，EEF 的过滤能力高于其他算法，从而减少了网络能耗。当 MAX_FALSH 设置为 2 时，由于允许途中节点继续传递虚假数据，消耗了网络能量，因此能量消耗高于 DEF 方案，但是仍低于 FIMA 和 SEF 方案。

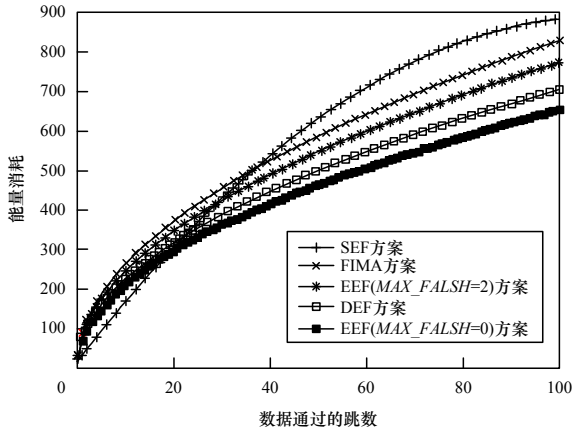

图 4 能量消耗比较

5 结束语

本文提出了一个无线传感器网络途中过滤增强方案，通过对途中节点使用安全增强方案，使过滤能力不因途中节点的受损而降低，从而提高了过滤能力，减少了虚假数据在网络中传递的次数。实验结果表明，该方案与 SEF、DEF、FIMA 相比节省了能量，从而延长了网络的生命期。

参考文献

[1] 杨 庚, 王江涛, 程宏兵, 等. 基于身份加密的无线传感器网络密钥分配方法[J]. 电子学报, 2007, 35(1): 180-184.

[2] 刘 帅, 林亚平, 余建平. 基于簇的传感器网络节点复制攻击检测[J]. 计算机仿真, 2007, 24(6): 129-132.

[3] Yu Lei, Li Jianzhong. Grouping-based Resilient Statistical En-route Filtering for Sensor Networks[C]//Proceedings of INFOCOM'09. Rio de Janeiro, Brazil: IEEE Press, 2009.

[4] Ye Fan, Luo Haiyun, Lu Songwu, et al. Statistical En-route Filtering of Injected False Date in Sensor Networks[J]. IEEE Journal on Selected Areas in Communications, 2005, 23(4): 839-850.

[5] Yu Zhen, Guan Yong. A Dynamic En-route Scheme for Filtering False Data Injection in Wireless Sensor Networks[C]//Proceedings of the 25th IEEE International Conference on Computer Communications. Barcelona, Spain: IEEE Press, 2006.

[6] Nghiem T P, Cho Tae-Ho. A Fuzzy-based Interleaved Multi-hop Authentication Scheme in Wireless Sensor Networks[J]. Journal of Parallel and Distributed Computing, 2009, 69(5): 441-450.

[7] Zhu Sencun, Setia S, Jajodia S, et al. An Interleaved Hop-by-hop Authentication Scheme for Filtering of Injected False Data in Sensor Networks[EB/OL]. [2012-04-22]. http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=1301328&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D1301328.

[8] 谢 婧, 李 曦, 杨 峰. 应对虚假数据注入结合途中过滤与溯源追踪方法[J]. 计算机系统应用, 2011, 20(12): 249-256.

[9] 杨 峰, 周学海, 张启元. 无线传感器网络高覆盖、低延迟途中过滤方法研究[J]. 计算机工程与科学, 2010, 32(11): 20-24.

[10] Karp B, Kung H T. Greedy Perimeter Stateless Routing for Wireless Networks[C]//Proceedings of the 6th Annual International Conference on Mobile Computing and Networking. New York, USA: ACM Press, 2000.

编辑 张 帆

(上接第 114 页)

[5] Alexander S. MD5 Considered Harmful Today[EB/OL]. [2011-11-20]. http://www.win.tue.nl/hashclash/rogue-ca.

[6] 孙小英. MD5 算法在 Web 数据库中的应用[J]. 软件导刊, 2009, 8(7): 56-57.

[7] 孙维国, 李浩然. MD5 算法在数据安全中的应用及安全性分析[J]. 微计算机应用, 2010, 31(10): 66-69.

[8] 张裔智, 赵 毅, 汤小兵. MD5 算法研究[J]. 计算机科学, 2008, 38(7): 295-297.

[9] 周 林, 王 政, 韩文报. MD5 差分和差分路径的自动化构造算法[J]. 四川大学学报: 工程科学版, 2010, 42(6): 133-137.

[10] 周 林, 韩文报, 祝卫华, 等. MDx 差分攻击算法改进及 GPGPU 上的有效实现[J]. 计算机学报, 2010, 33(7): 1177-1182.

[11] Wang Xiaoyun, Yu Hongbo. How to Break MD5 and Other Hash Functions[C]//Proc. of EUROCRYPT'05. Berlin, Germany: [s. n.], 2005: 19-35.

[12] 陈少晖, 翟晓宁, 阎 娜, 等. MD5 算法破译过程解析[J]. 计算机工程与应用, 2010, 46(19): 100-101.

[13] 黎 琳. Hash 函数 RIPEMD-128 和 HMAC-MD4 的安全性分析[D]. 济南: 山东大学, 2007.

[14] 陈士伟, 金晨辉. MD5 碰撞攻击的多重消息修改技术的研究[J]. 通信学报, 2009, 30(8): 89-95.

[15] 白洪欢. MD5 快速碰撞算法之研究[D]. 杭州: 浙江大学, 2010.

[16] 张绍兰, 刑国波, 杨义先. 对 MD5 的改进及其安全性分析[J]. 计算机应用, 2009, 29(4): 947-949.

[17] 崔国华, 周荣华, 粟 栗. 关于 MD5 强度分析的研究[J]. 计算机工程与科学, 2007, 29(1): 45-48.

[18] Ugmbbc. 可执行文件的 MD5 碰撞[EB/OL]. [2011-12-10]. http://www.cnbeta.com/articles/59117.htm.

[19] 陈士伟, 金晨辉. 对强化 MD 结构杂凑函数的一个新的"牧群"攻击[J]. 电子与信息学报, 2010, 32(8): 1953-1955.

编辑 张 帆