基于网络分簇和多路径的无线自组织网络匿名通信协议

张中科[1] 汪 芸[2]
(中国电子科技集团公司第38研究所 合肥 230088)[1]
(东南大学计算机科学与工程学院 教育部计算机网络和信息集成重点实验室 南京 210096)[2]

摘 要 提出了基于网络分簇和多路径的自组织网络匿名通信协议(CMAR),该协议中节点首先在不暴露身份信息的情况下,利用双线性配对技术生成和邻居节点共享的密钥,并以此为基础,秘密地建立用于成员节点和簇头节点之间通信的簇内路由表项,在簇内路由表项的辅助下,完成源节点和目标节点之间多路径的匿名建立,以及数据报文的匿名转发。通过性能分析可以发现,CMAR协议的密码学运算负荷较低,且网络通信性能良好。

关键词 无线自组织网络,网络分簇,多路径,匿名通信
中图法分类号 TP393.08 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.10.040

Cluster-based Multipath Anonymous Routing Protocol in Wireless Ad Hoc Networks

ZHANG Zhong-ke[1] WANG Yun[2]
(No. 38 Research Institute of CETC, Hefei 230088, China)[1]
(School of Computer Science & Engineering, Southeast University, Key Laboratory of CNII, MoE, Nanjing 210096, China)[2]

Abstract Without disclosing the real identities of participating nodes, the shared session keys and secret link identifiers among neighbors are exchanged based on bilinear pairing, and further the local anonymous routing entries between the normal nodes and cluster head are built. With the help of local anonymous routing table, multi-path between source-destination can be constructed and data packets will be anonymously forwarded to destination along these paths. Simulation results show that CMAR achieves perfect anonymous communication with relatively low cost of communication and computation overhead.

Keywords Wireless ad hoc network, Network clustering, Multi-path, Anonymous communication

1 引言

在现有的无线自组织网络通信协议中,通常节点身份ID或者位置信息会以明文方式出现在报文中,由于无线信号的暴露性和网络中节点性质的不确定性,报文极易被窃听和分析,从而造成节点相关信息的泄漏。节点隐私信息的泄漏会严重降低无线自组织网络的安全性和可用性,因为这些信息很容易被恶意攻击者利用来对网络实施进一步的破坏。

近些年来,设计无线自组织网络下安全高效的匿名通信协议成为了新的研究热点,已有相关文献从不同角度讨论了无线自组织网络下的匿名通信问题。ANODR[1]借助无线通信的广播特性,利用单向陷门函数和"回旋洋葱"等措施来实现匿名通信,但是该协议密码学计算开销大。MASK[2]应用双线性配对技术来实现邻居节点之间的相互匿名认证,形成共享的秘密链路标识和会话密钥,进而利用它们建立路由路径完成报文转发,但是它无法保护目标节点的ID信息。SPENA[3]以Sink为中心建立路由树,通过随机选择中间节点对报文重新加密来改变报文的内容,从而维护了源节点的隐私性;HANOR[4]提出了层次化的匿名通信路由协议;AOS[5]是一种匿名覆盖系统,但是它们难以适应自组织类型的网络。

针对现有研究工作的不足,本文提出了基于网络分簇和多路径的自组织网络匿名通信协议(CMAR),该协议中节点首先在不暴露身份信息的情况下,利用双线性配对技术生成和邻居节点共享的密钥,并以此为基础,秘密地建立用于成员节点和簇头节点之间通信的簇内路由表项,在簇内路由表项的辅助下,完成源节点和目标节点之间多路径的匿名建立,以及数据报文的匿名转发。

2 匿名路由协议设计

2.1 前提假设

假定每个节点分配了全局唯一的ID,ID是一定长度的位串,若节点ID的位串长度是128位,那么节点ID空间是2^{128},这是一个相当大的空间。本文采用类似文献[2]的伪名分配方案,即对任意节点ID_i,可以通过随机选择一个整数$r_A \in Z_q$,并通过$PS_{ij}=r_j \cdot PS_i$运算生成任意多的伪名,得到对应的$S_{ij}=r_j \cdot g \cdot H(PS_i)$。本文中每个节点的ID是秘密的,节点可以通过线下的方式获得它准备通信的节点$\langle ID_k$,

到稿日期:2013-12-08 返修日期:2014-02-14 本文受国家高技术研究发展计划("863"计划)基金资助项目(2011AA040502)资助。
张中科(1980—),男,博士,工程师,主要研究方向为网络安全、信息对抗,E-mail:zhang-zhongke@qq.com;汪 芸(1967—),女,博士,教授,主要研究方向为分布式计算、容错计算,E-mail:vunwang@seu.edu.cn(通信作者)。

PS_{kj}〉信息，其中 PS_{kj} 是节点 ID_k 的任意一个伪名，但是没有节点能够获得网络中所有节点的 ID 信息。

2.2 网络分簇

网络分簇的关键是选择簇头节点，簇头节点的选举旨在能够克服网络随机部署所带来的网络节点密度不均匀和局部拓扑差异的影响，簇头节点之间应具有一定的分离度，即，使得每个簇包含的节点数量相对均匀，每个簇的半径应该差异不大。文献[6]提出了无线自组织网络下的一种随机分布式的 landmark 节点选择方案，该方案不依赖于节点的地理位置信息，也不需要节点间进行时钟同步，具有简单性和实用性等特点，该方法同样可以用来在无线自组织网络中选择和部署簇头节点。

簇头节点选定后，它将向周边的邻居节点广播“分簇通告报文”，通告报文格式如下：(CH, Hop, TTL)，各字段分别代表簇头节点的 ID、距离簇头节点的跳数和报文生存期。簇头的邻居节点收到“通告报文”后，通过分簇通告报文，可以获知自己到邻近簇头节点的跳数距离，随后将“通告报文中”的 Hop 域加 1、TTL 值减 1，此时若 TTL 值为 0，则停止转发该报文，否则，继续转发该报文给自己的邻居节点。若节点同时收到多个簇的“分簇通告报文”，它将选择距离最近的簇加入；若它到两个簇头的距离相等，节点随机选择一个簇加入。

2.3 簇内路由表的建立

网络分簇完成后，簇头节点首先建立 Bloom Filter 数据结构 BF，BF 用于接收成员节点秘密传送过来的对应各自 ID 信息的多个 hash 值，并将 BF 中对应的位段置“1”，它可以帮助簇头节点判定路由请求报文中的目标节点是否位于簇内，减少不必要的密码学运算。本小节讨论建立辅助簇内路由的相关数据结构的问题，主要包括伪名广播，节点和邻居节点间的共享密钥和秘密链路标识的建立，以及簇间通信链路标识的建立等过程。

2.3.1 伪名广播

簇头节点发送“伪名通告报文”，报文的格式为 $(Flag, PS_{CH}, n_{CH}, PS_i, n_i, Hop)$，其中 $Flag$ 字段用来标识报文类型，此处是伪名通告报文；PS_{CH} 字段为簇头节点的伪名；n_{CH} 是簇头节点随机选定的随机数；PS_i 字段为转发“伪名通告报文”的簇内非簇头节点的伪名，簇头节点发送此报文时将其初值设为 NULL；n_i 字段是随机选定的随机数，簇头节点发送此报文时将其初值设为 NULL；Hop 字段为转发“伪名通告”报文的簇内非簇头节点到簇头节点的跳数，初值设为 0。

节点首次收到邻居节点发送的“伪名通告报文”时，将记录三元组 (PS_{CH}, Hop, n_i)，即簇头节点的伪名 PS_{CH}、自己到簇头节点的跳数距离和簇头节点选择的随机数；接着，检测 PS_i 字段是否为 NULL，若不为 NULL，则说明“伪名通告报文”是簇内非簇头邻居节点发送的“伪名通告报文”，此时节点记录三元组 (PS_i, Hop, n_i)，即邻居节点的伪名、邻居节点距离簇头的跳数和邻居节点选择的随机数。随后将 TTL 值减 1，将 Hop 值加 1，随机选择新的随机数 n_j，并在 PS_i 字段填上自己的伪名，发送自己的“伪名通告报文”。根据三元组 (PS_i, Hop, n_i)，节点将邻居节点分为 3 个集合：上行节点集、下行节点集，并行节点集。所谓“上行邻居节点”是指到簇头节点的 Hop 值比其小 1 的节点。所谓“并行邻居节点”是指到簇头节点的 Hop 值与其相等的节点。所谓“下行邻居节点”是指到簇头节点的 Hop 值比其大 1 的节点。

2.3.2 链路密钥和标识的建立

文献[2]提出了利用双线性配对来协商链路密钥和标识的方法，本文以该方法为工具创建用于匿名通信的簇内数据结构。伪名广播之后，节点运算自己和邻居节点的共享密钥，运算方法如下：$K_{ij}=K_{ji}=f(S_i, H(PS_j))=f(H(PS_i), S_j)=f(H(PS_i), H(PS_j))^g$。

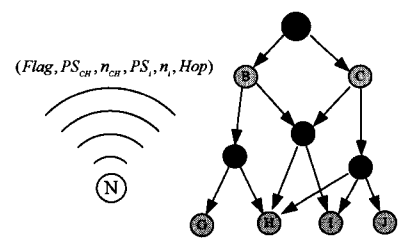
在此基础之上，节点运算自己和邻居节点之间共享的链路密钥和秘密的链路标识对 $(Key_{ij}^{\alpha}, LinkID_{ij}^{\alpha})$，方法如下：

$$Key_{ij}^{\alpha}=H'(K_{ij} \parallel n_i \parallel n_j \parallel 2\alpha)$$

$$LinkID_{ij}^{\alpha}=H'(K_{ij} \parallel n_i \parallel n_j \parallel 2\alpha+1)$$

通过选择不同的 α 值，节点 i 和节点 j 之间可以生成不同的链路密钥和链路标识。节点首先运算节点 $(Key_{ij}^{\alpha}, LinkID_{ij}^{\alpha})$ 作为初始的链路密钥和链路标识，节点还将提前运算 $(Key_{ij}^{\alpha+1}, LinkID_{ij}^{\alpha+1})$，并监听以 $LinkID_{ij}^{\alpha+1}$ 为链路标识的报文。在接下来的报文传输过程中，当节点 i 需要传输报文给节点 j 时，它将用链路标识取代目标节点的实际 ID。由于链路标识是在共享密钥 K_{ij} 的基础之上通过 hash 运算获得的，因此除了节点 i 和节点 j 之外，没有其它节点知道通信双方的真实身份，保证了通信的匿名性。

图 1 上部左图是广播报文的格式；上部右图描述了伪名广播过程，伪名广播从簇头节点开始，逐层向下进行；下图是节点 D 建立的与上下行、并行邻居节点之间的基本的密钥和秘密链路标识。但是仅用单一的 $LinkID_{ij}^{\alpha}$ 作为链路标识会造成节点无法决定怎样向下转发报文，以图 1 为例，当簇头节点 CH 有报文需要传送给成员节点 G 时，若仅用 $LinkID_{CHB}^{\alpha}$ 作为链路标识将报文传输到节点 B，节点 B 将无法判断报文是传送给自己还是下行节点 D、E。解决该问题的方法是：以 $(Key_{ij}^{\alpha}, LinkID_{ij}^{\alpha})$ 为基础，通过递增 α 值，派生出新的链路标识和密钥，从而用不同的链路标识来帮助成员节点决定报文的转发动作。



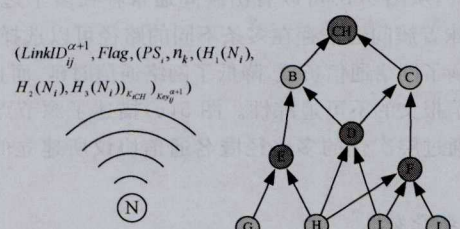
NodeID	Basic LinkIDs and Keys	
D	Up	$(LinkID_{DB}^{\alpha}, Key_{DB}^{\alpha}), (LinkID_{DC}^{\alpha}, Key_{DC}^{\alpha})$
	Par.	$(LinkID_{DE}^{\alpha}, Key_{DE}^{\alpha}), (LinkID_{DF}^{\alpha}, Key_{DF}^{\alpha})$
	Down	$(LinkID_{DH}^{\alpha}, Key_{DH}^{\alpha}), (LinkID_{DI}^{\alpha}, Key_{DI}^{\alpha})$

图 1 伪名广播过程

建立伪名广播和基础的簇内密钥、链路标识之后，簇内的非簇头节点将创建“链路标识生成通告报文”$(LinkID_{ij}^{\alpha+1}, Flag, (PS_i, n_k, (H_1(N_i), H_2(N_i), H_3(N_i))_{K_{iCH}})_{Key_{ij}^{\alpha+1}})$，其中 $LinkID_{ij}^{\alpha+1}$ 为节点和上行邻居集中的节点共享的秘密链路标识；$Flag$ 同样用来标识报文类型，此处是链路标识生成通

告报文；PS_i 为发送“链路标识生成通告报文”的节点的伪名；$H_1(N_i)$，$H_2(N_i)$，$H_3(N_i)$为 3 个不同的 hash 函数对节点 ID 进行散列运算后得到的值；n_k 为节点随机选定的随机数；K_{iCH}为节点和簇头节点共享的密钥；$Key_{ij}^{\alpha+1}$是与 $LinkID_{ij}^{\alpha+1}$ 对应的节点和其上行邻居节点共享的密钥。需要指出的是，在发送“链路标识生成通告报文”时，节点重新选择 r_j，并通过 $PS_{ij}=r_j \cdot PS_i$ 运算生成新的伪名及其对应的 $S_{ij}=r_j \cdot g \cdot H(PS_i)$，结合已保存的簇头节点的伪名 PS_{CH}，节点可以运算自己和簇头节点之间秘密的链路标识和链路密钥对(Key_{iCH}^{α}，$LinkID_{iCH}^{\alpha}$)。

节点在创建完成“链路标识生成通告报文”后，将报文发送给上行邻居节点集中的每个节点，发送“链路标识生成通告报文”时通过更新 α 值，运算出没有使用过的共享密钥和链路标识对，例如，若已使用过的最大 α 值的链路标识和共享密钥对为(Key_{ij}^{α}，$LinkID_{ij}^{\alpha}$)，则将 $LinkID_{ij}^{\alpha+1}$ 作为链路标识，用 $Key_{ij}^{\alpha+1}$来加密$(PS_i, n_k, (H_1(N_i), H_2(N_i), H_3(N_i))_{K_{iCH}})$。上行邻居节点集中的节点收到该报文后，同样更新标识和共享密钥；并将其转发给自己的上行邻居节点集中的每个节点，建立路由表项(*UpLinkID List*，*DownLinkID List*)对，其中，*UpLinkID List* 为转发报文给上行邻居节点集中节点时使用的链路标识，*DownLinkID List* 为下行节点发送报文给它时使用的链路标识。节点利用 $LinkID_{ij}^{\alpha+1}$ 对应的 $Key_{ij}^{\alpha+1}$ 可以解密$(PS_i, n_k, (H_1(N_i), H_2(N_i), H_3(N_i))_{K_{iCH}})$，得到创建“链路标识生成通告报文”节点的伪名 PS_i。如果节点先后收到多个下行邻居节点集中节点转发过来的含有相同的 PS_i 的“链路标识生成通告报文”，例如 M_1 和 M_2，则它只转发最先收到的报文 M_1，忽略后续收到的报文 M_2，并将 M_2 报文中的链路标识添加到转发 M_1 报文时建立的路由表项的 *DownLinkID List* 中。例如，图 2 中的节点 B，当节点 H 发送“链路标识生成通告报文”时，它将先后收到节点 D 和 E 转发过来的节点 H 的报文，但是它只转发其中的一个报文，同时路由表项 $LinkID_{BCH}^{\alpha+5}$ 对应两个下行链路标识 $LinkID_{BD}^{\alpha+2} \cdot LinkID_{BE}^{\alpha+3}$，当节点 B 收到以 $LinkID_{BCH}^{\alpha+5}$ 为链路标识的报文时，它随机选择 $LinkID_{BD}^{\alpha+2}$ 或者 $LinkID_{BE}^{\alpha+3}$ 作为报文转发的链路标识。按照上述方法，“下行链路生成通告报文”最终传递到簇头节点。



NodeID	UpLinkID	DownLinkID	Dest.
B	$LinkID_{BCH}^{\alpha+1}$	NULL	B
	$LinkID_{BCH}^{\alpha+2}$	$LinkID_{BD}^{\alpha+1}$	D
	$LinkID_{BCH}^{\alpha+3}$	$LinkID_{BE}^{\alpha+1}$	E
	$LinkID_{BCH}^{\alpha+4}$	$LinkID_{BE}^{\alpha+2}$	G
	$LinkID_{BCH}^{\alpha+5}$	$LinkID_{BD}^{\alpha+2}$，$LinkID_{BE}^{\alpha+3}$	H
	$LinkID_{BCH}^{\alpha+6}$	$LinkID_{BD}^{\alpha+3}$	I

图 2　簇内路由表的建立过程

簇头节点收到“链路标识生成通告报文”后，利用它和下行节点的共享的密钥解密得到$(PS_i, n_k, (H_1(N_i), H_2(N_i), H_3(N_i))_{K_{iCH}})$数据段。接着，簇头节点通过运算得到自己与伪名为 PS_i 节点的共享密钥 $K_{chi}=K_{ich}=f(S_{CH}, H(PS_i))$，并利用 K_{chi} 解密得到 $H_1(N_i)$，$H_2(N_i)$，$H_3(N_i)$，它将 BF 中 $H_1(N_i)$，$H_2(N_i)$，$H_3(N_i)$对应位加“1”，建立 $H_1(N_i)$，$H_2(N_i)$，$H_3(N_i)$和转发链路标识 *FwdLinkID List* 之间的对应关系，*FwdLinkID List* 包含了发送该“链路标识生成通告报文”给它的所有链路标识。

2.3.3　簇间链路密钥和标识的建立

若节点有邻居节点属于另外一个簇，则该节点为簇边界节点。例如，图 3 中左边簇的节点 A、B、C、D 和 E，右边簇的节点 a、b 和 c，都是簇边界节点。簇的边界节点 A 和 a 广播“伪名通告报文”后，它们之间的链路标识和链路密钥对(Key_{Aa}^{α}，$LinkID_{Aa}^{\alpha}$)均已建立，且获得了邻近簇的簇头节点的伪名 PS'_{CH}。此时簇边界节点将发送“簇间链路标识生成通告报文”，格式如下：$(LinkID_{ij}^{\beta}, Flag, (Seq, PS'_{CH}, n_k')_{Key_{ij}^{\beta}})$，其中，$LinkID_{ij}^{\beta}$ 为节点和上行邻居集中的节点最近没有使用过的共享的秘密链路标识，Key_{ij}^{β} 是与 $LinkID_{ij}^{\beta}$ 对应的节点和其上行邻居节点共享的密钥；$Flag$ 用来标识报文类型，此处是簇间链路标识生成通告报文；Seq 是随机选择的报文序列号，用来标识报文的唯一性，中间节点只转发具有同样 Seq 的“簇间链路标识生成通告报文”一次；PS'_{CH}，n_k'分别为邻接簇头节点的伪名和随机数。中间节点对于“簇间链路标识生成通告报文”的转发动作和“链路标识生成通告报文”的转发动作相同，该报文最终将传送至簇头节点。

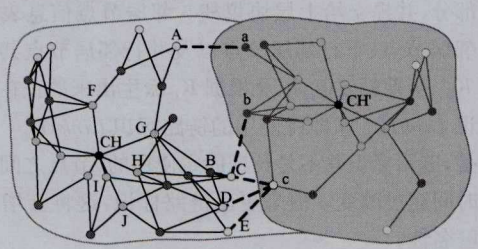
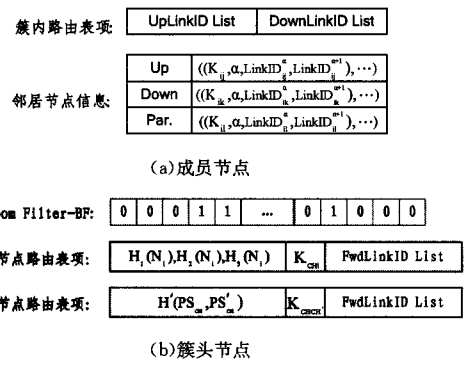


图 3　簇间路由相关数据结构的建立过程

簇头节点收到“簇间链路生成报文”后，通过 PS'_{CH}，n_k'可以运算出自己与邻接簇头节点之间共享的密钥 $K_{CHCH'}$。簇头节点以 $H'(PS_{CH}, PS'_{CH})$为邻接簇头的身份标识，同时建立 $H'(PS_{CH}, PS'_{CH})$和转发链路标识 *FwdLinkID List* 之间的对应关系。簇头节点可能会收到多个簇边界节点关于同一个邻接簇的“簇间链路标识生成通告报文”，例如，图 3 中的簇头节点 CH 会分别收到节点 A、B、C、D 和 E 的“簇间链路标识生成通告报文”，这些报文分别由邻居节点 F、G、H、I 和 J 转发过来，则与 $H'(PS_{CH}, PS'_{CH})$对应的 *FwdLinkID List* 包括了 $Link_{CHF}^{\beta}$，$Link_{CHG}^{\beta}$，$Link_{CHH}^{\beta}$，$Link_{CHI}^{\beta}$，$Link_{CHJ}^{\beta}$，当簇头节点 CH 需要发送报文给邻接簇头节点 CH'时，它将随机选择 F、G、H、I、J 中的任一节点转发。当报文传送到任一簇边界节点时，例如节点 A，它将以 $LinkID_{Aa}^{\alpha}$ 为链路标识将报文转发给节点 a，$LinkID_{Aa}^{\alpha}$ 是节点 a 和其它簇边界节点之间链路标识，每当簇边界节点收到其它簇边界节点转发过来的报文时，它将通过已建立的簇内链路标识将报文转发给自己所在簇的簇头节点，因此节点 a 会将报文传递给簇头节点 CH'。

2.3.4 簇内数据结构

通过上述过程，网络中的节点建立了相关数据结构，在网络中，由于簇头节点和成员节点的功能不一样，因此维护的数据结构也存在差异。图 4 分别列出了簇头节点和成员节点维护的数据结构。



(a)成员节点

(b)簇头节点

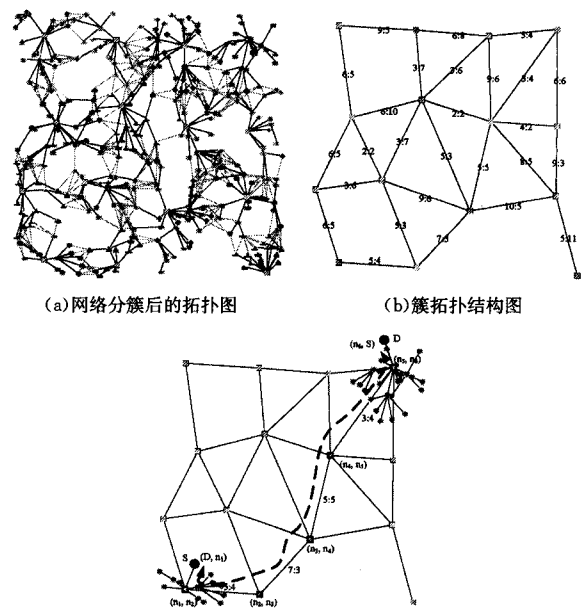
图 4 簇内节点所维护的数据结构

成员节点维护的数据结构有簇内路由表和邻居节点信息表，其中，簇内路由表主要用于报文在成员节点和簇头节点进行通信，每当节点收到带有 *UpLinkID List* 中的链路标识的报文时，成员节点将在 *DownLinkID List* 中随机选择链路标识，替换报文原有的链路标识，将报文转发出去；反之，若节点收到带有 *DownLinkID List* 中的链路标识的报文时，成员节点将在 *UpLinkID List* 中随机选择链路标识，替换报文原有的链路标识，将报文转发出去。若簇内路由表项中的 *DownLinkID List* 为空，则节点是报文的目标节点，此时解析报文的数据部分，并提交给上层协议栈。邻居节点信息表主要维护上行邻居节点、下行邻居节点以及并行邻居节点共享的基础密钥 K_{ij}、当前的 α 值，以及根据 K_{ij} 派生出来的正在使用的链路标识 $LinkID_{ij}^{\alpha}$ 和即将使用的链路标识 $LinkID_{ij}^{\alpha+1}$。需要指出的是，链路标识并不是长期固定使用的，节点之间可以每隔一定时间通过改变 α 值来更新链路标识，这将会增强网络通信的匿名性。

簇头节点维护的数据结构主要包括 Bloom Filter—BF、成员节点路由表和相邻簇头路由表，其中 BF 用来映射成员节点的 ID 信息，从而在网络路由过程中帮助簇头节点决定目标节点是否位于自己所在的簇内。在非静态的网络环境中，簇内的节点可能会发生变化，为了能够支持节点的动态加入和退出，可以考虑采用计数型 Bloom Filter 数据结构。成员节点路由表主要维护簇头节点和成员节点的共享密钥，以及转发报文给相应成员节点对应的链路标识列表。相邻簇头节点路由表主要维护簇头节点和相邻其它簇的簇头节点之间共享的密钥，以及对应的转发链路标识列表。需要指出的是，每当收到成员节点转发过来的“数据报文”时，簇头节点可以根据报文中的链路标识，在成员节点路由表项和相邻簇头节点路由表中反查出簇头节点和报文发送节点之间的共享密钥，从而可以利用该密钥解密得到报文的数据段。

2.4 匿名路由

簇与簇之间的关系图如图 5(b)所示，在图 5(b)中链路上的数字分别表示邻接的两个簇中位于簇边界节点的数量。如图 5(c)所示，层次化的多路径匿名通信协议分两个层次：簇内路由和簇间路由，首先源节点在簇内建立的秘密链路标识的保护下将路由请求发送给簇头节点，簇头节点利用 BF 检测目标节点是否在自己的簇内，若不在自己的簇内，则在簇间会话密钥的保护下将路由请求发送给邻近的簇头节点。同样，邻近的簇头节点也利用 BF 检测目标节点是否在自己的簇内，若目标节点位于自己所在的簇内，则在秘密链路标识的保护下将路由请求转发给目标节点，否则继续簇间路由，直到路由请求转发到目标节点所在簇头节点为止，最后由目标节点所在簇的簇头节点通过已建立的簇内路由表将报文转发到目标节点。在此过程中，目标节点的识别有别于以往的匿名通信路由协议，以往匿名通信路由协议的路由请求在泛洪的过程中，收到泛洪报文的每个中间节点都需要利用共享的对称密钥或者公钥密码体制下的私钥解密报文来获知自己是否是路由请求的目标节点。而本文所提的匿名通信协议由于簇头秘密地维护了映射成员节点 ID 的 Bloom Filter，仅通过查询 Bloom Filter 就可以决定路由请求报文的去向，因此非目标节点所在簇内的节点无需进行不必要的密码学运算，降低了节点的运算开销。



(a)网络分簇后的拓扑图　(b)簇拓扑结构图

(c)基于簇拓扑图的路径发现

图 5 匿名路由示意图

另外，从图 5(b)可以看出簇间通常存在多个边界节点，这也意味着簇间通信存在多条不同的路径可以选择，多路径不但平衡了网络通信负载，降低了网络通信时延，而且增强了网络通信报文的不可追踪性。图 5(c)描述了源节点 S 和目标节点通过层次化的多路径匿名通信协议所建立的通信路径。

2.4.1 路径发现

层次化的多路径匿名通信协议的路由分为两个层次：簇内路由和簇间路由，路由表项以链路标识取代节点 ID，具体包括如下步骤：

Step 1(路由请求生成和簇内路由)　当源节点 S 需要与目标 D 通信时，它将生成 RREQ 报文发送至簇头节点，RREQ 报文格式如下：$(LinkID_{Sj}^{\alpha+1}, Flag, (RREQ, PS_S, Seq, (D,r)_{K_{SD}}, n_k, H_1(D), H_2(D), H_3(D))_{K_{SCH}})$，其中，$LinkID_{Sj}^{\alpha+1}$ 是源节点和上行邻居节点集中的节点共享的链路标识；

$Flag$ 为报文类型标识，此处为“数据报文”类型；$(RREQ, PS_S, Seq, (D,r)_{K_{SD}}, n_k, H_1(D), H_2(D), H_3(D))_{K_{SCH}}$ 为报文数据段；$RREQ$ 表示报文路由请求报文。PS_S 为源节点的伪名；Seq 为随机生成的路由请求报文的序列值，(PS_S, Seq) 标识着报文的唯一性；$(D,r)_{K_{SD}}$ 是用源和目标节点共享密钥对目标节点 ID 和随机数 r 进行加密后得到的；n_k 为源节点随机选定和簇头节点共享的会话标识；$H_1(D), H_2(D), H_3(D)$ 为用 3 个不同 hash 函数对目标节点 ID 进行散列后得到的值；K_{SCH} 为源节点和簇头节点共享的密钥。

利用簇内已建立的数据结构，该报文将会被路由至簇头节点，报文在簇内转发时，链路标识会根据节点的本地路由表项进行替换，其它内容不会发生变化。簇头节点收到报文时，根据和簇内上一跳节点共享的链路标识 $LinkID_{K_{CH}}^{i+1}$，在“成员节点路由表”中反查出其与源节点之间的共享密钥 K_{SCH}，并用 K_{SCH} 解密得到数据段，在 BF 中用 $H_1(D), H_2(D), H_3(D)$ 检测目标节点是否位于自己所在的簇内。若目标节点位于簇内，则通过 $H_1(D), H_2(D), H_3(D)$ 查找获得目标节点对应的转发链路标识，随机选定新的会话标识 n_k'，并用自己和目标节点共享的密钥 K_{CHD} 加密数据段，将报文传送给目标节点；若目标节点不在簇内，则将 RREQ 报文中的数据段用与邻接簇头节点共享的密钥加密后，更新会话标识为 n_k'，将 RREQ 报文转发给邻接簇的簇头节点。

Step 2(簇间路由)　邻居簇头节点收到 RREQ 报文后，同样首先根据链路标识在本地的“相邻簇头节点路由表”中反查出自己与前一簇头节点之间共享的密钥，解密得到数据段，随后检测 (PS_S, Seq)，判断以前是否收到过同样伪名和序列值的报文，若收到过，则丢弃该报文；否则，解密数据段，更新会话标识为 n_k'，建立 (n_k, n_k') 之间的对应关系，图 5(c)举例说明了源节点 S 和目标节点 D 之间路径建立后，相关节点上的路由表项。同样邻居簇头节点在 BF 中利用 $H_1(D), H_2(D), H_3(D)$ 检测目标节点是否位于自己所在的簇内，若在，则用与目标节点共享的密钥加密数据段，将 RREQ 报文转发给成员节点；否则，用与邻接簇头节点共享的密钥加密数据段，将报文转发给邻接簇头节点。从上述过程可以发现，RREQ 报文在簇间路由是以“最先到达”为转发准则，而不是根据“跳数最小”来决定是否转发，对于 RREQ 报文的路由转发采用“最先到达”策略和“跳数最小”策略均不会产生环路。本文之所以采用“最先到达”策略，主要是因为同样是相邻的两个簇，若以节点为单位来计算跳数往往存在很大差异，因此在簇间采用“跳数最小”往往获得的不是源和目标节点间的最短路径，相反采用“最先到达”原则选择的路径更符合网络负载的现实状况，且通常接近最短路径。

Step 3(路由回复)　目标节点收到路由请求报文后，利用和簇头节点共享的密钥 K_{CHD} 解密数据段得到源节点的伪名 PS_S，运算与源节点的共享密钥 $K_{SD}=K_{DS}=f(PS_S, S_D)$，并用该密钥解密 $(D,r)_{K_{SD}}$，以验证自己是否为目标节点。若验证通过，目标节点运算 $(D,r+1)_{K_{SD}}$ 用于向源节点认证自己，接着生成 RREP 报文，报文格式如下：$(LinkID_{D_j}^{i+1}, Flag, ((RREP, D, r+1)_{K_{SD}}, n_k)_{K_{DCH}})$，其中，$LinkID_{D_j}^{i+1}$ 为目标节点和上行节点共享的秘密链路标识；$Flag$ 字段标识报文类型，此处为数据报文；RREP 表示报文为路由回复报文。会话标识 n_k 为簇头节点转发 RREQ 报文给目标节点时所使用的值；该报文的数据段将用目标节点和其簇头节点的共享密钥 K_{DCH} 加密。

目标节点首先将 RREP 报文发送给簇头节点，簇头节点收到报文后，查找路由表中与 n_k 对应的 n_k'，并用 n_k' 替换报文中的 n_k，通过 n_k' 对应的 $FwdPointer$ 查找到与上一跳簇簇头节点共享的密钥以及转发链路标识，重新加密数据段，将报文转发给上一跳簇头节点。该过程一直继续下去，直到报文传递到源节点所在的簇头节点位置，报文最终按照 RREQ 报文传输的路径原路返回到达源节点。源节点收到报文后，利用共享密钥验证目标节点，若验证通过，则源和目标节点之间的通信路径建立完成，最终建立的“数据路由”表项如图 6 所示，其中簇头节点 $FwdPointer$ 是指向簇内路由表项或者相邻簇头节点间路由表项的指针，通过它可以查找到相应的转发链路标识和共享密钥。

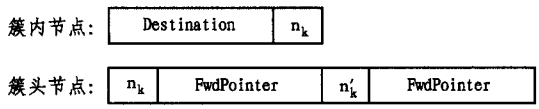


图 6　路径建立后的路由表项

2.4.2　数据报文转发

当源节点 S 有报文需要发送给目标节点 D 时，源节点 S 首先查找路由表项获得目标节点 D 对应的会话标识 n_k，随后生成数据报文 $(LinkID_{S_j}^{i+1}, Flag, (Data, n_k, Payload)_{K_{SCH}})$，此处 $Flag$ 字段为“数据报文”类型；$Data$ 表示报文类型是数据报文；n_k 是源节点和其所在簇的簇头节点间共享的会话标识；$Payload$ 是报文的信息部分。数据报文的数据段将用源节点 S 和簇头节点的共享密钥加密，利用已建立的簇内路由表项将其发送到簇头节点。

簇头节点收到报文后，在簇内路由表中反查出与源节点共享的会话密钥 K_{SCH}，利用与源节点 S 共享的密钥 K_{SCH} 解密得到数据段，根据会话标识 n_k 在“数据路由表”中查找到对应的 n_k'，用 n_k' 替换数据段中的 n_k，通过 n_k' 的 $FwdPointer$ 指向的路由表项查找到会话和下一簇簇头节点或者目标节点的会话密钥，重新加密报文的数据段，最后在 $FwdPointer$ 指向的路由表项的 $FwdLinkID\ List$ 中随机选择链路标识，将报文转发过去。上述过程一直重复，直到报文到达目标节点为止。

2.5　多路径效果

在数据报文的传输过程中，路径存在多样性。首先，源节点或目标节点到其所在簇的簇头节点存在多路径，通过上述簇内路由表的建立过程可以发现，若成员节点有多个上行邻居节点，或者成员节点到簇头节点的路径上的任一节点有多个上行邻居节点，则该节点到簇头节点存在多条通信路径。其次，簇与簇之间也存在多路径，因为，相邻簇之间通常有多个边界节点，边界节点到簇头存在多路径，相邻簇的边界节点之间也存在多路径。这种路径的多样性结合链路标识的秘密性，增加了追踪分析数据报文的难度，提高了通信的匿名性。

3　性能分析

3.1　密码学运算负荷分析

ANODR 和 MASK 是两种典型的匿名通信协议，ANO-

DR 在通信节点对之间建立的是单路径，MASK 则在通信节点对之间建立了链路不相交的多路径，很多其它的匿名通信协议都由它们变化而来。从密码学运算的角度来看，这些 ANODR 或者 MASK 的改进协议与改进前的协议差别不大，只不过为了达到匿名通信的目标，所采用的具体手段不一样。本节将对比分析 CMAR 与 ANODR、MASK 在密码学运算负荷方面的性能差异。这 3 种协议分别涉及到了安全散列技术、对称密钥密码学技术、基于 ECC 的非对称密钥密码学技术和双线性配对技术，其中最耗时的是后两者，本文所指的非对称密钥加解密均指 ECC 下的运算，因为相对于运算负荷重的 RSA 算法，在低端物理设备上 ECC 算法更具有可行性。

参考文献[9-11]，表 1 给出了不同平台下密码学运算的执行时间，这些数据均是在物理节点上通过实验取得的。相比后两者，安全散列和对称密钥加解密速度要快得多，文献[12]给出了它们的实验数据，根据不同的处理器，数值在1～25Mb/s 之间不等。在下面的分析过程中，假定网络中的节点采用的处理器主频为 100～200MHz，结合表 1 中的数据可以计算出不同密码学操作对应的时间。

表 1　不同平台下密码学运算的执行时间

不同平台下 ECC 算法实现的执行时间			
节点	Mica2	Tmote Sky	Yopy
CPU	8bit Atmega128L	16bit MSP430	32bit StrongARM
主频	7.3728MHz	8MHz	200MHz
加密	1.96s	1.12s	0.0465s
解密	1.35s	0.77s	0.0245s
不同平台下双线性配对算法实现的执行时间			
节点	Mica2	Tmote Sky	Imote2
CPU	8bit Atmega128L	16bit MSP430	32bit PXA271
主频	7.3728MHz	8MHz	104MHz
时间	1.90s	1.27s	0.06s

CMAR 和 MASK 均有初始数据结构的建立过程，每个节点通过双线性配对运算获得它与邻居节点以及簇头节点之间的共享密钥，双线性配对运算次数等于邻居节点数量加 1，但是该运算频度较低，节点之间主要通过改变 α 值重新执行 hash 运算来更新链路标识和链路密钥，因此所带来的运算负荷并不大。MASK 在路由发现过程中不需要进行密码学运算，不过 MASK 协议不能保护目标节点的 ID 信息，在数据报文转发过程中，为了防止攻击者通过比较报文的内容来追踪数据流动的路径，MASK 要求所有中间节点运用对称密钥对数据报文进行重新加密和解密，每次非对称密钥运算耗时 0.04ms。ANODR 的路由请求 RREQ 报文泛洪时，RREQ 报文经过的节点都需要耗时 0.02ms 对报文进行一次对称密钥的加密操作，为了判断自己是否是路由请求报文的目标节点，若目标节点的 ID 信息是用非对称密钥加密，这些节点还需要耗时 24.5ms 执行非对称密钥解密操作，此外还要耗时 50ms 左右生成一对临时使用的用于保护路由表建立的非对称密钥。ANODR 在路由回复报文的转发过程中，反向路径上的节点不但需要耗时 0.02ms 执行一次对称密钥解密操作，还需耗时 24.5+46.5=71ms 执行一次非对称密钥的加解密操作。在数据报文转发过程中，类似于 MASK，ANODR 要求所有中间节点耗时 0.04ms 运用对称密钥对数据报文进行重新加密和解密。CMAR 中只有路由请求 RREQ 报文经过的中间簇头节点需要耗时 0.04ms 执行一次对称密钥的加解密操作，报文到达目标节点后，目标节点需要耗时 60ms 执行一次双线性配对运算以获取和源节点之间的共享密钥。CMAR 中路由回复报文沿反向路径返回源节点时，同样只有反向路径上的中间簇头节点需要耗时 0.04ms 执行一次对称密钥的加解密操作。在数据报文的转发过程中，CMAR 也只需要路径上的每个中间簇头节点耗时 0.04ms 执行一次对称密钥的加解密操作。此外，需要特别指出的是，ANODR 的路由请求报文是全网泛洪的，而 CMAR 的路由请求报文只在簇头节点间泛洪，因此总量上执行密码学运算的节点要少得多。

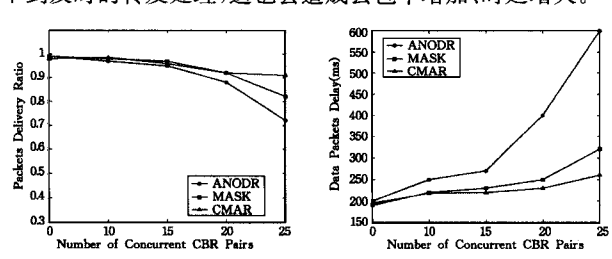
通过上述分析不难发现，ANODR 的密码学运算负荷最大，MASK 密码学运算负荷较小，但是会付出暴露目标节点 ID 的代价，相比较而言，CMAR 只付出了较小的密码学运算代价就保护了节点的身份信息完全匿名。

3.2　通信性能分析

仿真实验以 NS2 为实验平台，信道和无线模型为 Two-ray ground reflection model，MAC 层采用 IEEE 802.11 DCF (Distributed Coordination Function)，节点的通信范围为 250 米，无线信道的带宽为 2Mbps，在 2000×2000 的场景下部署了 400 个节点，网络分簇数为 15，平均簇大小约为 28。模拟实验中并行活跃的会话数量从 5 到 25 不等，会话的源节点和目标节点在网络中随机选取，背景流量由 CBR 产生，传输层协议采用 UDP，源节点发送的数据报文的大小为 512 字节，源节点的报文在网络层发送的平均速率为每秒 4 个报文，相继发送的报文时间间隔满足指数分布。参照表 1 中的数据，本节保守假定节点的对称密钥运算能力为 1Mb/s，则 512 字节的报文由于对称密钥加解密运算造成的时延为 1ms。

本节主要从两个方面对比分析三者在数据传输性能方面的差异：(1)数据报文投递率：目标节点接收到的报文数量 b 和源节点发送的报文数量 a 之间的比例 b/a；(2)数据报文传输时延：网络中传输的所有报文的平均时间间隔 (t_1, t_2)，其中 t_1 是指源节点发送报文的时间，t_2 是指目标节点接收到报文的时间。

从图 7 可以发现，随着网络背景流量的增加，三者的报文投递率均有所降低，传输时延有所增加，相比较而言，CMAR、MASK 的通信性能明显优于 ANODR。主要是因为 ANODR 在通信节点对之间建立的是单路径，而 CMAR、MASK 建立的是多路径，因此网络负载更加均衡，出现网络拥塞和缓冲区队列溢出的几率和程度均要略低于 ANODR。另外一方面，ANODR 的密码学运算量较大，随着网络流量的增加，报文得不到及时的转发处理，这也会造成丢包率增加、时延增大。



(a)报文投递率　　(b)报文传输时延

图 7　CMAR 和 ANODR、MASK 在报文投递率方面的性能差异

对 CMAR 和 MASK 进行对比，可以发现，在网络背景流量不大的情况下，CMAR 和 MASK 性能相近，MASK 甚至会优于 CMAR，这是因为 MASK 密码学运算负荷比 CMAR 低；但是随着网络背景流量增加，CMAR 的性能明显比 MASK 要好，这得益于 CMAR 的分簇多路径相对于 MASK 简单的节点不相交多路径方式，无论是多路径的条数还是路径的分散度、报文经过路径的随机性均有很大优势，因此网络负载更加均衡。

结束语 本文首先基于双线性配对技术，描述了在分簇结构的网络中邻居节点之间密钥协商以及簇内路由表项建立的过程。在此基础之上，本文设计了基于网络分簇和多路径的匿名通信路由协议(CMAR)，该协议不但能够达到全面的匿名通信的目标，减小匿名通信所带来的密码学运算代价，而且充分利用了簇内和簇间存在多路径的优势，增强了网络通信能力和抗报文的追踪性能力。通过理论分析和仿真实验可以发现，CMAR 协议密码学运算负荷较小且通信性能良好。

参 考 文 献

[1] Kong J, Hong X. ANODR: Anonymous on Demand Routing with Untraceable routes for Mobile Ad-hoc Networks[C]// Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing (MOBIHOC'03). 2003:291-302

[2] Zhang Y, Liu W. MASK: Anonymous On-demand Routing in Mobile Ad Hoc Networks[J]. IEEE Transactions on Wireless Communications, 2006, 5(9):2376-2385

[3] Pongaliu K, Xiao L. Maintaining Source Privacy under Eavesdropping and Node Compromise Attacks[C]// Proceedings of the 30th IEEE International Conference on Computer Communications(INFOCOM'11). 2008:1656-1664

[4] Liu J, Hong X, Kong J, et al. A Hierarchical anonymous Routing Scheme for Mobile Ad-Hoc Networks[C]// Proceedings of the 2006 IEEE conference on Military communications (MILCOM'06). 2006:2310-2316

[5] Zhang R, Zhang Y, Fang Y. AOS: An Anonymous Overlay System for Mobile. Ad hoc Networks[J]. Wireless Networks, 2011, 17(4):843-859

[6] Nguyen A, Milosavljevic N, Fang Q, et al. Landmark Selection and Greedy Landmark-descent Routing for Sensor Networks[C]// Proceedings of the 26th IEEE Conference on Computer Communications(INFOCOM' 07). 2007:661-669

[7] Shao M, Hu W. Cross-layer Enhanced Source Location Privacy in Sensor Networks[C]// Proceedings of IEEE Conference on Sensor, Mesh & Ad Hoc Communication Networks (SECON' 09). 2009:1-9

[8] Wang H D, Sheng B, Li Q. Privacy-aware Routing in Sensor Networks[J]. Computer Networks, 2009, 53(9):1512-1529

[9] Wang H, Sheng B, Tan C, et al. WM-ECC: an elliptic curve cryptography suite on sensor motes [R]. Technical Report WM-CS-2007-11. CS Department, College of William and Mary, 2007:1-14

[10] Oliveira L, Aranha D, Gouvêa C, et al. TinyPBC: Pairings for authenticated identity-based non-interactive key distribution in sensor networks [J]. Computer Communications, 2011, 34(3):485-493

[11] Szczechowiak P, Kargl A, Scott M, et al. On the application of pairing based cryptography to wireless sensor networks[C]// Proceedings of the 2nd ACM Conference on Wireless Network Security(WISEC'09). 2009:1-12

[12] Ganesan P, Venugopalan R, Peddabachagari P, et al. Analyzing and modeling encryption overhead for sensor network nodes[C]// Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications(WSNA'03). 2003:151-159

(上接第 168 页)

[7] http://www.datacenterknowledge.com/archives/2012/03/14/estimate-amazon-cloud-backed-by-450000-servers

[8] Zhu X S. *Research on Semantic Peer-to-peer Overlay Route Model*[J]. Computer Engineering, 2008, 43(13):110-112

[9] Zhang Y J, Gu J H, Wang X Z. A Hierarchical P2P Semantic Overlay Network Architecture Based on Topic and Physical Proximity[J]. Journal of Electronics & Information Technology, 2008, 30(8)

[10] Wang C Z, Yang N, Chen H W. Improving Lookup Performance Based on Kademlia[C]// Proc. of the Second International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC 2010). Hubei: 2010:446-449

[11] Lian Q, Chen W, Zhang Z. On the Impact of Replica Placement to the Reliability of Distributed Brick Storage Systems[C]// Proceedings 25th IEEE International Conference on Distributed Computing Systems, 2005(ICDCS 2005). 2005:187-196

[12] Chand R, Cosnard M, Liquori L. Powerful resource discovery for Arigatoni overlay network [J]. Future Generation Computer Systems, 2008, 24(1):31-38

[13] Stevens T, Wauters T, Develder C, et al. Analysis of an anycast based overlay system for scalable service discovery and execution[J]. Computer Networks, 2010, 54(1):97-111

[14] Wu W M, Wu Y J, Zhao W Y. Chord-based Semantic Web Service Discovery[J]. Acta Electronica Sinica, 2007, 35(B12):152-155

[15] Zhang Y, Huang H, Yang D, et al. Bring QoS to P2P-based semantic service discovery for the Universal Network[J]. Personal and Ubiquitous Computing, 2009, 13(7):471-477

[16] Di Stefano A, Morana G, Zito D. A P2P strategy for QoS discovery and SLA negotiation in Grid environment[J]. Future Generation Computer Systems, 2009, 25(8):862-875

[17] Zhou J, Dou W. A QoS-Aware Service Selection Approach on P2P Network for Dynamic Cross-Organizational Workflow Development[C]// Proc. of the International Conference on Web Information Systems and Mining (WISM 2009). Shanghai: Springer-Verlag Berlin, 2009:289-298