一种新的基于上下文传递的临近空间安全切换机制

徐国愚 陈性元 杜学绘
(解放军信息工程大学 郑州 450001)

摘 要 针对临近空间中安全切换问题,提出一种新的基于上下文传递的安全切换机制。首先,设计了一种面向临近空间浮空器的切换基站预测算法,以基于多普勒频移技术计算飞行器发生切换的时间与位置,确定切换基站;其次,利用上下文传递机制预先将认证信息发送给切换基站,保证切换过程中通信的可靠性。性能分析与仿真实验表明,该机制通信与计算开销小,强制中断概率低,能够满足临近空间的应用需求。

关键词 临近空间、安全切换、上下文传递机制、多普勒频移
中图法分类号 TP393.08 **文献标识码** A

New Near Space Security Handoff Scheme Based on Context Transfer

XU Guo-yu CHEN Xing-yuan DU Xue-hui
(The PLA Information Engineering University, Zhengzhou 450001, China)

Abstract To solve the problem of security handoff in near space, a new security handoff scheme based on context transfer was proposed. Firstly, a handoff destination estimate algorithm was designed, which is based on the Doppler shift mechanism that can estimate the handoff time and location. Secondly, the previous base-station sent authentication message to the next based-station in advance based on context transfer, which can increase handoff efficiency. Performance analysis and simulation results show that the communication and computation overhead of the scheme are small, and the forced termination probability is low. The scheme is suitable for the application of near space.

Keywords Near space, Security handoff, Context transfer, Doppler shift

1 前言

临近空间(Near Space)是指距离海平面20～100千米的区域,位于空基和天基之间。在临近空间中浮空器平台对地相对静止,具有覆盖面广、飞行持续性强、侦查分辨率高、战场存活性高等特点。临近空间浮空器通过连接空基、天基及地基,能够提供空天地一体化网络,在军事通信保障、导弹防御以及自然灾害监控等领域都具有十分重大的意义[1,2]。

在临近空间通信中,由于高速飞行器的动态性,存在着切换问题。如图1所示,高速飞行器通过临近空间浮空器基站能够接入空天地一体化网络,通过路由中转与后方指挥中心进行实时通信。但是当其飞行至浮空器通信覆盖区域边缘时,需要将通信链路切换到下一跳浮空器或者卫星基站。为保证切换过程中的安全与高效性,文献[3,4]提出一种基于上下文传递的临近空间安全切换机制,该机制基于切换概率权值矩阵推算出可能切换的基站集合,通过预先传递安全信息给下一跳切换基站,避免了重复执行接入认证流程,但是文献没有给出获取切换概率的具体方法。文献[5]提出了一种基于预认证的临近空间安全切换机制,但是该方法交互次数多、计算量大,且由于空天链路的大时延及高中断性,该方法容易造成高失败率;另外,该文献同样未给出具体的切换基站选择算法。

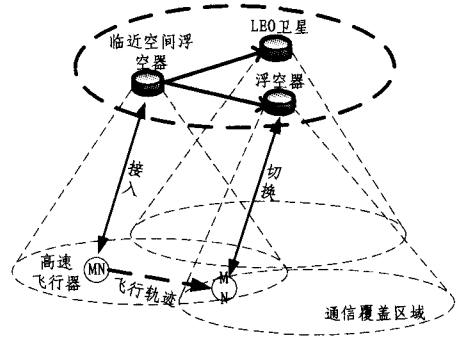

图1 临近空间飞行器切换场景

目前尚未见到专门针对临近空间平台的切换基站选择算法研究。文献[6,7]针对LEO卫星网络中地面节点切换问题,提出一种基于动态多普勒频移技术的LEO切换选择算法。文献[8]在其基础上提出了一种面向飞行器的LEO卫星切换选择算法。但是临近空间浮空器对地相对静止,因此无法直接应用上述卫星切换机制。

针对现有问题,本文提出一种新的基于上下文的临近空间安全切换机制。首先在第2节,借鉴文献[6-8]的方法,设计了一种面向临近空间浮空器的切换基站预测算法,即利用动态多普勒频移技术,计算出飞行器节点发生切换的时间及

到稿日期:2012-11-04 返修日期:2013-02-02 本文受国家973重点基础研究发展计划(2011CB311801),河南省科技创新人才计划(114200510001)资助。

徐国愚(1982－),男,博士生,主要研究领域为无线网络安全;陈性元(1963－),男,博士,教授,主要研究领域为网络安全技术;杜学绘(1968－),女,博士,教授,主要研究领域为网络安全技术。

位置信息，预测出下一跳切换基站；第 3 节给出了基于上下文传递机制的临近空间安全切换机制，即当前基站根据切换预测信息及切换策略选择下一跳切换基站，预先将安全上下文信息传递给切换基站，以提高切换效率；第 4、5 节进行协议分析与仿真验证；最后总结全文。

2 基于多普勒频移技术的临近空间切换基站预测算法

当高速飞行器接入到临近空间浮空器基站后，浮空器基站开始计算飞行器发生切换的时间与位置，以确定下一跳切换基站。其中，切换时间的计算方法与文献[6,8]类似。另外，为便于分析，假设高速飞行器处于巡航阶段，飞行速度、方向及高度保持不变。

2.1 高速飞行器切换时间计算

首先给出临近空间浮空器与高速飞行器节点之间地心角的计算公式。如图 2 所示，A 为临近空间浮空器，海拔高度为 H；B 为高速飞行器，海拔高度为 h；R_E 为地球半径。浮空器通过检测多普勒频移，可以得到飞行器与浮空器的通信仰角 E，因此能够计算出地心角，如式(1)所示。

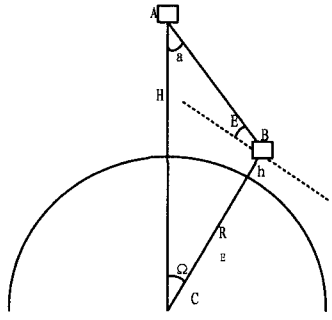

图 2 浮空器与飞行器的地心角计算

$$\begin{aligned}&\frac{\sin(a)}{R_E+h}=\frac{\sin(90^\circ+E)}{R_E+H}\\&\frac{\sin(90^\circ-\Omega-E)}{R_E+h}=\frac{\sin(90^\circ+E)}{R_E+H}\\&\Omega=\arccos\left(\frac{R_E+h}{R_E+H}\cos(E)\right)-E\end{aligned}\tag{1}$$

以地心为球心、R_E+h 为半径做一个球面，将浮空器的通信覆盖区域投影到该球面上，如图 3 所示。令 N 为浮空器在该球面上的映射点，在 t_0 时刻，飞行器位于 A 点，t_1 时刻飞行至 B 点，能够通过式(1)计算出角距离 $AN=\arccos\left(\frac{R_E+h}{R_E+H}\cos(E_0)\right)-E_0$，$BN=\arccos\left(\frac{R_E+h}{R_E+H}\cos(E_1)\right)-E_1$。$E_0$，$E_1$ 为相应的通信仰角。

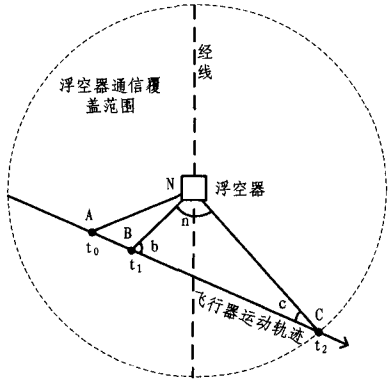

图 3 高速飞行器切换时间计算

通过飞行器的角速度 v 可以计算出角 $AB=v\cdot(t_1-t_0)$。对球面三角型 ANB 运用余弦定理可以获得角 $b=180^\circ-\arccos\left(\frac{\cos(AN)-\cos(AB)\cos(BN)}{\sin(AB)\sin(BN)}\right)$。设飞行器与浮空器的最小通信仰角为 $E_{\min}$，则能够计算出 $CN=\arccos\left(\frac{R_E+h}{R_E+H}\cos(E_{\min})\right)-E_{\min}$。利用正弦定理可以获得方位角 $c=\arcsin\left(\frac{\sin(BN)\sin(b)}{\sin(CN)}\right)$，角 $n=180^\circ-b-c$。最终通过等式 $BC=v\cdot(t_2-t_1)=\arcsin\left(\frac{\sin(CN)\sin(n)}{\sin(b)}\right)$，可以计算出节点发生切换的时间 $t_2=\frac{1}{v}\cdot\arcsin\left(\frac{\sin(CN)\sin(n)}{\sin(b)}\right)+t_1$。

2.2 高速飞行器切换位置计算

在图 4 中，点 N 为临近空间浮空器的位置，C 为高速飞行器发生切换时的位置，$\overrightarrow{DC}$为飞行器的航向线。由于临近空间浮空器对地静止，能够通过其所在的经纬度 NB、BE 计算出 $NE=\arccos(\cos(NB)\cdot\cos(BE))$。对应地，能够计算出角 $E=\arcsin\left(\frac{\sin(NB)}{\sin(NE)}\right)$，角 $N_1=90^\circ-E$。根据飞行器的航向角可以获得航向线与经线的逆时针夹角 d，因此可以获得角 $N_2=180^\circ-d-c$，其中 c 为飞行器发生切换时的方位角(c 的计算见 2.1 节)。最终可以获得角 $N=N_1+N_2$。另外，可以计算出 $CE=\arccos(\cos(CN)\cos(NE)+\sin(CN)\sin(NE)\cos(N))$(其中 CN 的计算见 2.1 节)。通过正弦定理，可以获得 $E_1=\arcsin\left(\frac{\sin(CN)\sin(N)}{\sin(CE)}\right)$，$E_2=E-E_1$。最终可以计算出 $CA=\arcsin(\sin(CE)\cdot\sin(E_2))$，$AE=\arccos(\cos(CA)\cos(CE)+\sin(CA)\sin(CE)\cos(90^\circ-E_2))$。

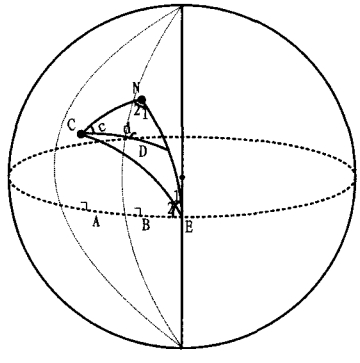

图 4 高速飞行器切换地理坐标计算

3 基于上下文传递的临近空间安全切换机制

本文基于上下文传递方法[9]给出临近空间安全切换机制，通过在认证实体间预先传递认证信息，减少认证对切换性能的影响，其具体过程如图 5 所示。

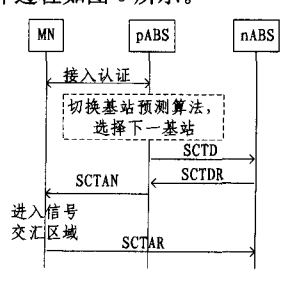

图 5 基于上下文传递的临近空间安全切换机制流程

1. 高速飞行器节点 MN 首先与浮空器基站进行接入认证，并协商出会话密钥 SK_0；同时，MN 节点还会将相关的切

换策略(例如最大通信时间优先或者最小通信距离优先等策略)发送给浮空器基站。

2. 浮空器节点作为先前接入基站(Previous Access Base Station,pABS),采用第 2 节的切换预测算法计算出 *MN* 节点发生切换的时间 T 及坐标,根据 *MN* 节点的切换策略确定下一跳切换基站(Next Access Base Station,nABS)。

3. 在时刻 $T-T_{th}$ 时,pABS 将当前 *MN* 节点的安全上下文信息 $SCT=(ID_{MN},EA,AA,T,SK_1)$ 封装成 SCTD 消息发送给 nABS,其中 T_{th} 为发送上下文信息的切换门限值,ID_{MN} 代表切换节点标识,EA,AA 代表通信加密及认证算法,T 代表预期切换时间,$SK_1=PRF(SK_0,R_0)$ 为切换会话密钥,PRF 为伪随机数生成函数,R_0 为随机数,由 pABS 产生并对外保密。为保证 SCTD 消息传输时的安全性,pABS 采用文献[10]基于身份的签密算法对 SCTD 消息进行签名与加密。

4. nABS 收到 SCTD 消息后,将安全上下文信息 *SCT* 保存,并根据相关切换策略为 *MN* 节点预留信道资源等。之后,nABS 返回成功应答消息 SCTDR,SCTDR 消息同样使用签密算法进行安全性保护。

5. pABS 节点收到 SCTDR 消息后,发送安全上下文通告消息 SCTAN 给 *MN* 节点,SCTAN 消息中包含有 nABS 节点的标识、会话密钥 SK_1 等内容,SCTAN 消息通过 *MN* 与 pABS 的会话密钥 SK_0 进行加密与认证,*MN* 节点收到后将相关消息保存。

6. 当飞行器 *MN* 到达 pABS 与 nABS 的通信覆盖区域交汇处时,节点 *MN* 发送切换请求命令 SCTAR。SCTAR 消息中包含有 *MN* 节点、pABS 及 nABS 的标识、当前时间戳 T_{MN} 等信息。SCTAR 消息通过切换会话密钥 SK_1 进行加密与认证。

7. nABS 节点收到 SCTAR 消息后,首先获取当前时间戳 T_{cur},验证 $|T_{cur}-T_{MN}|\leqslant\Delta t_1$,$\Delta t_1$ 为系统门限值,若正确,则查找上下文信息,获取会话密钥 SK_1,解密并验证 SCTAR 消息,完成整个切换过程。另外,nABS 节点若在预期切换时间 $T+\Delta t_2$(Δt_2 为门限值)到达后仍未收到 SCTAR 消息,将删除对应的安全上下文信息,释放预留通信资源。

为保证切换的安全性,*MN* 节点在经过多次切换,或者切换时间到达预定门限时,需要重新进行接入认证。接入认证过程可以与传输并发进行,保证了通信过程不会受到重认证的影响。

4 机制性能与安全性分析

下面对本机制的性能与安全性进行分析,并与现有机制进行对比。由于文献[3]仅给出了切换协议框架,未涉及具体实现细节,因此下面主要与文献[5]的机制进行比较。

4.1 性能分析

(1)本文提出了一种基于多普勒频移的临近空间切换预测算法,其能够根据浮空器基站的特点,提高切换预测的精度;而文献[5]并未给出具体的切换预测算法。

(2)本文协议通过 4 次上下文消息传递、4 次双线性对运算即可完成切换;而文献[5]至少需要传输 6 次消息、8 次双线性对运算,通信与计算开销较大。

(3)pABS 发送给 nABS 的 SCTD 消息的安全性通过基于身份的签密算法实现,不需要 pABS 与 nABS 提前建立安全通道,也不需 pABS 在线实时查询 nABS 的公钥,因此在保证安全的同时,降低了通信时延。

4.2 安全性分析

本机制通过会话密钥保证 *MN* 与 pABS 及 nABS 消息传输的安全性。会话密钥通过 *PRF* 函数派生,*PRF* 函数的单向性保证了后续 ABS 基站无法获取先前的会话密钥。同时,在会话密钥生成过程中添加随机数 R,能够保证先前 ABS 无法通过自身所拥有的会话密钥推导出后续会话密钥。例如,基站 nABS 派生出会话密钥 $SK_2=PRF(SK_1,R_1)$ 发送给下一跳基站 nABS′,由于随机数 R_1 仅被 nABS 基站知道,保证了 pABS 仅能通过 SK_0 获取 SK_1,但是无法获取 SK_2 及以后的会话密钥。

5 仿真验证

本文使用 OPNET 仿真工具对本机制及文献[5]的性能进行验证与比较,仿真主要关注切换时的强制通信中断概率 P_f,即某个基站内因切换失败导致中断的呼叫数与总的呼叫请求数之间的比值。仿真初始参数如表 1 所列,其中最大驻留时间 T_s 表示飞行器在浮空器覆盖区域内驻留的最长时间。仿真过程中,浮空器基站组成接入网络,飞行器随机接入网络,并以设定的速度和高度直线飞行。

表 1 仿真参数初始设置

参数	数值
浮空器/飞行器数目	50/250
浮空器/飞行器高度(km)	30/20
浮空器对飞行器通信覆盖半径(km)	40
高速飞行器巡航速度 V_a(km/s)	0.34
最大驻留时间 T_s(min)	3.92
呼叫到达率(10^{-4} calls/sec)	1.1
呼叫时长(s)	180
仿真时间(s)	3600

首先,仿真验证不同切换门限 T_{th}(即当前基站预先发送上下文信息给切换基站的时间)与浮空器通信负载(OPNET 网络背景流量)对本文机制的强制中断概率 P_f 的影响,如图 6 所示。从图中可以看出,当切换门限与驻留时间之比(T_{th}/T_s)大于 0.3 时,本文的机制受浮空器通信负载的影响较小,能够保证强制中断概率 P_f 小于 1%,其主要原因是,当 $T_{th}/T_s>0.3$ 时,能够保证切换基站有充足的时间完成对节点的预先认证,并预留出通信信道,避免了节点切换到下一基站时因为认证时延或者通信资源不足而被迫中断通信。

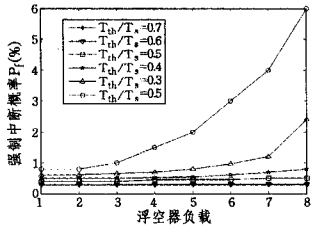

图 6 不同通信负载情况下对 P_f 的影响

图 7 为不同飞行器飞行速度 V_a 条件下,本机制与文献[5]机制的 P_f 比较,令文献[5]采用与本文相同的切换预测算法,$T_{th}/T_s=0.4$。从图中可以看出,当飞行器速度增大时,本文的 P_f 值基本不受影响,其主要原因是本机制的通信与计算开销较小,切换时间短,而文献[5]由于通信与计算开销较大,易受链路时延及误码率的影响,因此强制中断概率受影响较大。

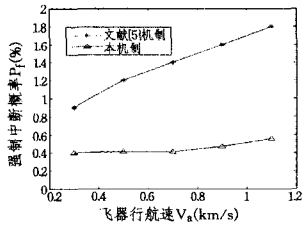

图 7　不同飞行器速度条件下对 P_f 的影响

通过仿真实验可以看出，本文提出的安全切换机制的强制中断概率不受基站通信负载及接入节点飞行速度的影响，能够适应临近空间应用场景。

结束语　本文提出了一种面向临近空间浮空器基站的安全切换机制。通过基于多普勒频移的切换基站预测算法，确定切换基站；通过上下文机制预先将节点认证信息与会话密钥传递给切换基站，避免了切换过程中的认证时延与通信中断。最后，通过性能分析与仿真实验证明了本机制的可靠性。

参 考 文 献

[1] Tomme L C E B, Phil D. The Paradigm Shift to Effects-Based Space: Near-Space as a Combat Space Effects Enabler[R]. Airpower research institute, 2005

[2] 聂万胜，罗世彬，丰松江，等. 近空间飞行器关键技术及其发展趋势分析[J]. 国防科技大学学报，2012，34(2)：107-113

[3] 钱雁斌，陈性元，杜学绘. 临近空间网络安全切换机制研究[J]. 计算机工程与应用，2008，44(15). 18-21

[4] Qian Yan-bin, Chen Xing-yuan, Du Xue-hui. A Security Context Transfer Method for Integrated space network[C]//2008 International Symposium on Information Science and Engieering. 2008：276-280

[5] 彭长艳. 空间网络安全关键技术研究[D]. 长沙：国防科学技术大学，2010

[6] Papapetrou E, Pavlidou R N. QoS handover management in LEO/MEO satellite systems[J]. IEEE Transactions on Communications, 2003, 46(3)：309-313

[7] Papapetrou E, Pavlidou F-N. Analytic Study of Doppler-based Handover Management in LEO Satellite Systems [J]. IEEE Transactions on Aerospace and Electronic Systems, 2005, 41(3)：830-839

[8] 陈炳才，韩亚萍，郭黎利，等. 低轨卫星网络支持飞机用户的切换管理算法[J]. 计算机应用，2009，29(8)

[9] Loughney J, Nakhjiri M, Perkins C, et al. Context tranfer protocol[M]. Internet-Draft, August 2004

[10] Jin Zheng-ping, Zuo Hui-juan, Du Hong-zhen, et al. An Efficient and Provably-Secure Identity-Based Signcryption Scheme for Multiple PKGs[C] // Proceedings of the International Conference on Computer Science and Information Technology. Singapore, 2008：189-193

(上接第 141 页)

SysBench 多线程测试运行如下：

```
sysbench --test=threads --num-threads=256
    --thread-yields=100 --thread-locks=2 run
```

参数分别表示创建 256 个线程；每个线程运行“加锁-运行-释放锁”过程 100 次；创建互斥锁 2 个。整个运行过程总时间(total time)在多次测试的情况下平均为 13.34s。

VTOS 的线程调度算法在设计上使得同进程中的线程尽量被集中在局部时间内进行调度运行，这在一定程度上可以减少系统运行时的进程切换次数。因为尽可能地使得前后两次在同一处理器上运行的线程属于同一个进程，这样的线程切换并不涉及到页表 CR3 控制寄存器的更改，从而也就避免了一次 TLB 硬件高速缓存刷新，提高了系统的运行效率。

结束语　本文分析了微内核架构多线程机制的研究现状和存在的问题，提出了一个微内核线程对象分层模型，包括基本功效层、实现层和优化层，并采用形式化的方式对各层进行了描述。在此基础上，本文描述了线程安全机制，包括进程空间隔离性、线程栈空间隔离性、线程资源共享和寄存器隔离性。这一对象模型和安全机制的形式化描述将作为后续的安全性形式化验证的基础。同时，本文设计了多线程机制的线程间通信、调度和互斥同步方案。本文描述和设计的微内核架构多线程机制在微内核操作系统 VTOS 上得以实现，并进行了功能和性能测试。实验结果表明，其有效地实现了 VTOS 的多线程机制，并具有很好的系统性能。

接下来的工作计划将是多线程机制的形式化描述转换为后续的安全性形式化验证；同时，将 VTOS 的多线程机制与多核处理有效地结合，研究新的调度算法，并提供灵活的多线程与多核管理工具。

参 考 文 献

[1] Zhou D. Towards the Formal Modeling of a Secure Operating System [C]// Proc. 23rd National Information System Security Conference. 2000

[2] Liedtke J. On μ-Kernel Construction [C]// Proc. 15th ACM symposium on Operating Systems Principles (SOSP'95). 1995：237-250

[3] Heiser G, Elphinstone K, Kuz I, et al. Towards trustworthy computing systems: taking microkernels to the next level [J]. ACM SIGOPS Operating Systems Review, 2007, 41(4)：3-11

[4] CMU CS. Mach Project [EB/OL]. http://www-2.cs.cmu.edu/afs/cs/project/mach/public/www/mach.html, 2011-06-05

[5] Accetta M, Baron R, Bolosky W, et al. Mach: A New Kernel Foundation for UNIX Development [J]. Computer, 1986, 39(4864)：1-16

[6] L4 Group. The L4 μ-Kernel Family [EB/OL]. http://os.inf.tu-dresden.de/L4/

[7] Elkaduwe D, Klein G, Elphinstone K. Verified Protection Model of the seL4 Microkernel [C]//LNCS 5295. 2008：99-114

[8] Klein G, Andronick J, Elphinstone K, et al. seL4: Formal Verification of an Operating-System Kernel [J]. Communications of the ACM, 2010, 53(6)：107-115

[9] Smullyan R M. First-Order Logic (Dover Books on Mathematics) [M]. Mineola: Dover Publications, 1995

[10] Stallings W. Operating Systems: Internals and Design Principles [M]. New Jersey: Prentice Hall, 2004

[11] Bovet D P, Cesati M. Understanding the Linux Kernel (3rd) [M]. Sebastopol: O'Reilly, 2005

[12] Franke H, Russell R, Kirkwood M. Fuss, futexes and furwocks: Fast Userlevel Locking in Linux [C]// Proc. Ottawa Linux symposium. 2002