doi:10.3969/j.issn.1001-0505.2015.01.005

基于扩频 Manchester 码的可靠自同步网络隐蔽时间通信模型

郭晓军[1,2,3] 程 光[1,3] 周爱平[1,3] 潘吴斌[1,3] 朱琛刚[1,3]

([1] 东南大学计算机科学与工程学院, 南京 210096)
([2] 西藏民族学院信息工程学院, 咸阳 712082)
([3] 东南大学计算机网络和信息集成教育部重点实验室, 南京 210096)

摘要: 针对包间延迟网络隐蔽时间信道存在的鲁棒性差、同步机制脆弱问题,提出了一种基于 Manchester 编码的可靠自同步网络隐蔽时间通信模型. 首先,对秘密消息进行扩频操作,得到扩频码. 然后,将流持续时间划分为若干相同长度时隙,每相邻两时隙构成一对,通过调整时隙对内包数量来模拟扩频码对应的 Manchester 编码中 0 和 1 的编码过程,以实现扩频码在流中的嵌入. 同时,采用时间偏移量指示同步位置,使得调制后的流呈现自同步性,以便接收端准确恢复秘密消息. 实验结果表明,与包间延迟方法相比,该模型能使收发双方更快速准确地保持同步,在不同网络负载下,秘密消息检测错误率最大值降低约 85%,显著提升了对网络干扰因素的抵抗能力,且在网络流量较大时呈现出更好的隐蔽性.

关键词: 信息安全;网络隐蔽时间信道;Manchester 编码;鲁棒性;隐蔽性
中图分类号: TP393 **文献标志码**: A **文章编号**: 1001-0505(2015)01-0023-08

Robust and self-synchronous network covert timing communication model based on spread Manchester code

Guo Xiaojun[1,2,3] Cheng Guang[1,3] Zhou Aiping[1,3] Pan Wubin[1,3] Zhu Chengang[1,3]

([1] School of Computer Science and Engineering, Southeast University, Nanjing 210096, China)
([2] School of Information Engineering, Tibet Nationalities Institute, Xianyang 712082, China)
([3] Key Laboratory of Computer Network and Information Integration of Ministry of Education, Southeast University, Nanjing 210096, China)

Abstract: To solve the problem of poor robustness and vulnerable synchronization of the current network covert timing channel using inter-packet delay (IPD), a robust and self-synchronous covert timing communication model based on spread manchester code (ROSMC) is proposed. First, the spectrum of the covert message is spread to produce spreading code (SC). Then, the duration of network flow is divided into many time intervals with equal length and two adjacent time intervals constitute one pair. Each bit of SC is embedded into flow through simulating 0 and 1 encoding process of the corresponding Manchester code (MC). The encoding process can be implemented by adjusting the number of the packets in the time interval pair. Meanwhile, an offset from the starting moment of flow is used to indicate the synchronous position. The offset and MC features make the adjusted flow present self-synchronization which can help receiver decode covert message accurately. The experimental results show that, compared with the IPD-based methods, the proposed model synchronizes sender and receiver more quickly and accurately. The maximum of the detection error rate can be reduced by 85% under different network traffic loads and the resistance to interference is significantly enhanced. Besides, the proposed model presents better covertness under heavier network traffic loads.

Key words: information security; network covert timing channel; Manchester code; robustness; covertness

收稿日期: 2014-07-20. **作者简介**: 郭晓军(1983—),男,博士生;程光(联系人),男,博士,教授,博士生导师,gcheng@njnet.edu.cn.
基金项目: 江苏省未来网络前瞻性基金资助项目(BY2013095-5-03)、江苏省普通高校研究生科研创新计划资助项目(KYLX_0141)、西藏自治区自然科学基金资助项目(2013).
引用本文: 郭晓军,程光,周爱平,等. 基于扩频 Manchester 码的可靠自同步网络隐蔽时间通信模型[J]. 东南大学学报:自然科学版,2015,45(1):23-30. [doi:10.3969/j.issn.1001-0505.2015.01.005]

Internet 发展已步入云计算和大数据时代,在经济利益驱动下,以用户信息泄露为代表的网络信息安全问题日益突出[1-2]. 作为威胁网络信息安全的有效手段之一,网络隐蔽通信技术利用计算机网络中公开合法信道进行秘密信息传输,其本质是通过改变网络流量中的相关特征来实现信息隐藏,已成为攻击者绕开网络安全策略发布攻击命令、窃取隐私数据等行为的重要途径. 目前,网络隐蔽通信技术分为存储信道[3-4]、行为信道[5]和时间信道[6-10]3 类. 其中,时间信道能顺利穿越网络中间设备,实用性较强,已受到国内外学者广泛关注.

现有的网络隐蔽时间信道主要通过改变载体流内单个 IPD 来嵌入并传输秘密消息. Cabuk 等[6]提出了一种隐秘时间信道,通过在固定时间内是否发送数据包来表示 0 和 1,使载体流的 IPD 呈现出 2 种不同长度;但此载体流存在明显统计规律,难逃统计方法检测,易暴露秘密消息. Archibald 等[7]在收发端借用 Luby-Transform 喷泉码,并引入防护频带方式,在 IPD 调制幅度与信道隐蔽性之间取得平衡;但其本质仍是对单个 IPD 进行改变,易受网络丢包、延迟抖动等因素影响而破坏秘密消息,鲁棒性较差. 针对此缺陷,钱玉文等[8]设计了一种基于 HTTP 协议的网络隐蔽时间信道,利用 HTTP 协议 POST 与 GET 的双工方式来提高可靠性;但该方法仅用单个较大 IPD 来保持收发端同步,一旦 IPD 被破坏,会造成收发端同步过程失败和信道瘫痪,且该信道不能用于 UDP 流. 为改善同步机制,牛小鹏等[9]采用网络隐蔽存储信道携带特殊标记来同步收发端;但含特殊标记的数据包发生重传、乱序或丢失[10-11]时,此同步机制会被破坏,从而导致秘密消息传输失败[12].

针对当前基于单 IPD 网络隐蔽时间信道存在的鲁棒性差、同步机制脆弱等问题,本文提出了一种基于 Manchester 编码(MC)的自同步网络隐蔽时间通信模型 ROSMC. 给出了关键算法、同步机制及相关参数设置的具体实现过程,并对其同步性、鲁棒性及隐蔽性进行了实验验证.

1 网络隐蔽时间通信模型

ROSMC 模型由发送端、秘密消息编码器(MC encoder)、解码器(MC decoder)及接收端组成(见图 1). 发送端产生正常载体流 f 后,调用消息编码器对秘密消息进行扩频处理得到扩频码;然后根据扩频码及同步参数,通过调整流 f 的时间特征来模拟扩频码所对应的 MC 序列中 0 与 1 的编码过程,从而完成秘密消息和同步信号的嵌入. 流 f 经网络传输后变成流 f'. 接收端调用解码器对捕获的流 f' 执行解扩和解码操作,以恢复流 f' 所携带的秘密消息,从而完成隐蔽通信过程.

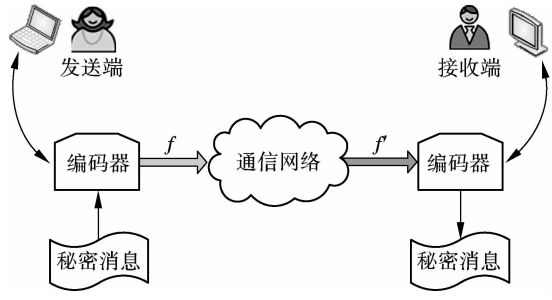
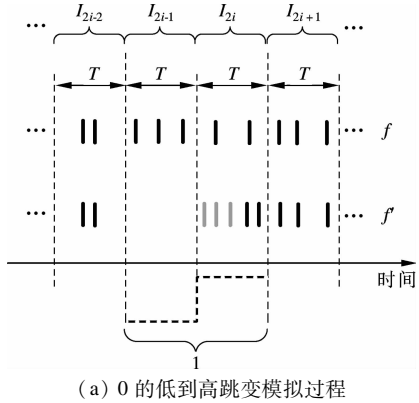


图 1 ROSMC 模型

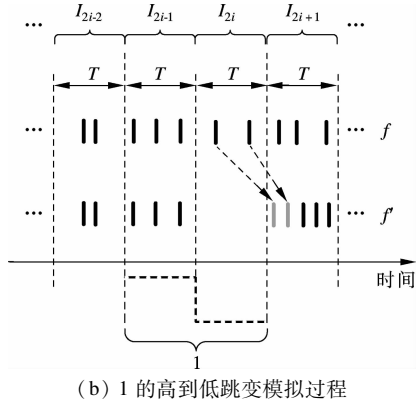
1.1 秘密消息编码器的编码过程

MC 是一种同步时钟编码技术,在物理层被用于编码同步位流的时钟和数据. 经 MC 编码后的数据位中间具有高低电平跳变,该跳变可同时作为同步时钟信号与数据信号;例如,可用低到高跳变表示 0,高到低跳变表示 1. 收发方可从 MC 信号序列中根据跳变信息来提取同步信息及数据.

本文借鉴 MC 思想,通过调整流 f 的时间特征来模拟 MC 编码过程,以实现秘密消息在流 f 中的嵌入. 此过程的关键之处在于如何模拟 MC 中 0 和 1 的高低电平跳变. 本文通过零操作(OPRT-0)和壹操作(OPRT-1)来实现(见图 2). 图中,深色和浅



(a) 0 的低到高跳变模拟过程



(b) 1 的高到低跳变模拟过程

图 2 OPRT-0 和 OPRT-1 过程

色条带分别表示时隙位置未被改变和已被改变的数据包. 详细过程为:从距流 f 起始时刻偏移 $\theta(\theta>0)$ 处起,选取一段持续时间 U,将 U 划分为 $2n(n>0)$ 个长度为 $T(T>0)$ 的时隙 $I_1,I_2,\cdots,I_{2n}$,且每 2 个相邻时隙构成 1 个时隙对 $(I_{2i-1},I_{2i})(i=1,2,\cdots,n)$. OPRT-0 表示对 (I_{2i-1},I_{2i}) 执行操作 Empty (I_{2i-1},T),使得 $A_{2i-1}=0,A_{2i}>0$,以模拟 MC 中 0 的低到高电平跳变;OPRT-1 表示对 (I_{2i-1},I_{2i}) 执行操作 Empty (I_{2i},T),使得 $A_{2i-1}>0,A_{2i}=0$,以模拟 MC 中 1 的高到低电平跳变. 其中,$A_1,A_2,\cdots,A_{2n}$ 表示落在各时隙内的包数量. Empty (I_x,T) 表示清空 $I_x(x=2i$ 或 $x=2i+1)$ 内所有数据包,即对 I_x 内每个包增加延迟,使其推迟到下一个时隙 I_{x+1} 内发送;其伪代码如算法 1 所示,算法的时间与空间复杂度分别为 $O(A_x+A_{x+1})$ 和 $O(1)$.

算法 1　时隙 I_x 内数据包清空操作

输入:I_x,T.

输出:I_x 内包延迟后的发送时刻.

```
Empty(I_x,T)
    a←C[ ]  /* C[ ]记录各时隙中第 1 个数据包在 Q[ ]处的索引 */
    b←C[x+1]
    Δ=T/(A[x]+A[x+1]+1)
    for s=0 to A[x] do
        t_p=xT+(s+1)Δ
        Q[a+s]=t_p
    end for
    for s=0 to A[x+1] do
        t_p=xT+(A[x]+s+1)Δ
        Q[b+s]=t_p
    end for
    A[x+1]=A[x]+A[x+1]    /* 将 I_x 所有包加入到 I_{x+1}中 */
    C[x+1]=C[x]
```

为提高 ROSMC 模型对网络时延、丢包、重传、抖动等因素的抵抗能力(即鲁棒性),本文采用直接序列扩频机制对秘密消息进行扩频处理[13],以得到扩频码,然后据此对流 f 进行 OPRT-0 和 OPRT-1 操作. 扩频处理过程为:设秘密消息 $\boldsymbol{M}=\{m_1,m_2,\cdots,m_n\}^{\mathrm{T}}$ 为二进制串,且 $|\boldsymbol{M}|=n$,$m_i\in\{0,1\}$. 令 $\boldsymbol{S}$ 为 PN 码,根据下式将 $\boldsymbol{M}$ 扩频为 $\boldsymbol{M}^{\mathrm{D}}$:

$$\boldsymbol{M}^{\mathrm{D}}=\boldsymbol{M}\times\boldsymbol{S}=\begin{pmatrix}m_1\\m_2\\\vdots\\m_n\end{pmatrix}\times\boldsymbol{S}=\begin{pmatrix}\boldsymbol{m}_1^{\mathrm{D}}\\\boldsymbol{m}_2^{\mathrm{D}}\\\vdots\\\boldsymbol{m}_n^{\mathrm{D}}\end{pmatrix}\tag{1}$$

$$\boldsymbol{S}=\begin{cases}\boldsymbol{S}_0 & m_i=0\\ \boldsymbol{S}_1 & m_i=1\end{cases}\tag{2}$$

$$\boldsymbol{S}_0=\{\alpha_1,\alpha_2,\cdots,\alpha_r\},\quad \boldsymbol{S}_1=\{\beta_1,\beta_2,\cdots,\beta_r\}$$
$$\alpha_k,\beta_k\in\{0,1\}\qquad k=1,2,\cdots,r\tag{3}$$

式中,$|\boldsymbol{S}_0|=|\boldsymbol{S}_1|=r$,其中 $r(r>0)$ 为扩频因子. 则 m_i 扩频后为 $\boldsymbol{m}_i^{D}$,即

$$\boldsymbol{m}_i^{\mathrm{D}}=\begin{cases}\{m_i\alpha_1,m_i\alpha_2,\cdots,m_i\alpha_r\} & m_i=0\\ \{m_i\beta_1,m_i\beta_2,\cdots,m_i\beta_r\} & m_i=1\end{cases}\tag{4}$$

式中,$|\boldsymbol{m}_i^{\mathrm{D}}|=r$.

为简单起见,本文设 $\alpha_1=\alpha_2=\cdots=\alpha_r=0,\beta_1=\beta_2=\cdots=\beta_r=1$. 例如,当 $\boldsymbol{M}=\{m_1,m_2\}=\{0,1\}$,$r=3$ 时,$\boldsymbol{S}_0=\{\alpha_1,\alpha_2,\alpha_3\}=\{0,0,0\}$,$\boldsymbol{S}_1=\{\beta_1,\beta_2,\beta_3\}=\{1,1,1\}$,则 m_1,m_2 扩频后分别为 $\boldsymbol{m}_1^{\mathrm{D}}=\{0,0,0\}$,$\boldsymbol{m}_2^{\mathrm{D}}=\{1,1,1\}$,因此扩频后 $\boldsymbol{M}^{\mathrm{D}}=\{0,0,0,1,1,1\}$,如图 3 所示.

秘密消息编码器在流 f 中嵌入 $\boldsymbol{M}$ 的过程伪代码如算法 2 所示.

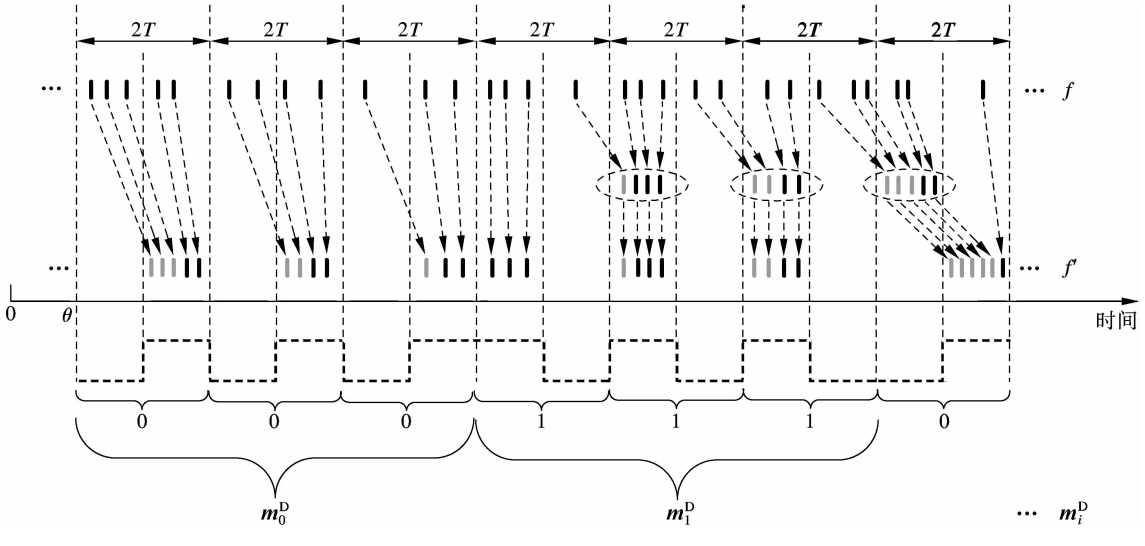


图 3　基于 MC 编码过程($r=3$)

算法 2 秘密消息编码器对 $\boldsymbol{M}$ 的嵌入

```
输入:M,θ,r,T.
输出:调制后的载体流 f.
Encode(M,θ,r,T)
    n←Len(M)
    M^D = DSSS(M,r)      /* 对 M 进行扩频操作 */
    G←2nr      /* G 为时隙总数 */
    L←record the duration time of flow f
    Q[ ]←record the timestamp of each packet in f
    if (2Tnr<(L-θ))
    {t_p←get the first timestamp in Q[ ] which >θ
      for i=0 to G-1 do
        A[i]=0;
        while (t_p>θ+iT && t_p<θ+(i+1)T) do
          if (A[i]==0)
            C[i]←record the position of first t_p in I_i;
          end if
            A[i]++      /* I_i 内包数量 */
            t_p←get next timestamp in Q[ ]
        end while
      end for
      for i=0 to nr-1 do      /* 对 M^D 的每个比特编码 */
          if (M^D[i]==0)
            Empty(I_{2i-1},T)
          else
            Empty(I_{2i},T)
          end if
      end for
    send all packets of flow f using new t_p in Q[ ]
    }
    else
      printf("Can't encode M with current flow f!")
    end if
```

收发双方在隐蔽通信前可事先通过其他安全途径获得私密参数(θ,T,r,U). 该算法的时间复杂度最坏情况为 $O(G^2)$,空间复杂度为 $O(|Q[\,]|)$,其中 $|Q[\,]|$ 为 $Q[\,]$ 的长度.

1.2 解码器的解码过程

解码器为秘密消息编码器的逆过程,可利用参数(θ,T,r,U)对携带 $\boldsymbol{M}$ 的流 f' 进行解码与解扩,并恢复出秘密消息 $\boldsymbol{M}$. 详细步骤如下:

① 根据参数 U,T,r,计算 $\boldsymbol{M}$ 的长度 n 和时隙总数 G.

② 当流 f' 的第 1 个数据包到达后,等待 θ,以获取 $\boldsymbol{M}$ 在流 f' 中的起始位置.

③ 统计从 θ 开始的 $I_1 \sim I_G$ 时隙内的数据包数量,保存在 $A[0]\sim A[G-1]$ 中.

④ 比较第 i 个时隙对(I_{2i},I_{2i+1})内的数据包数量. 若 $A[2i]-A[2i+1]<0$,则认为(I_{2i},I_{2i+1})对应的比特位为 0,即 $\boldsymbol{M}^{\mathrm{D}}$ 中第 k 个比特位为 0;否则为 1. 重复此过程以解码出 $\boldsymbol{M}^{\mathrm{D}}$.

⑤ 由 $\boldsymbol{M}^{\mathrm{D}}$ 解扩出 $\boldsymbol{M}$. 在 $\boldsymbol{M}^{\mathrm{D}}$ 中找到 m_i 所对应的 r 个扩频位,并保存在 $R[1]\sim R[r]$ 中. 为简单起见,令 $q=R[1]+R[2]+\cdots+R[r]$,若 $q=0$,说明 $R[1]=R[2]=\cdots=R[r]=0$,则 $m_i=0$;若 $q=r$,说明 $R[1]=R[2]=\cdots=R[r]=1$,则 $m_i=1$;若 $q\neq 0$,则解扩结果错误;重复此过程直至解扩出 $\boldsymbol{M}$.

⑥ 返回恢复出的秘密消息 $\boldsymbol{M}$.

上述步骤对应算法的时间、空间复杂度与算法 2 类似.

1.3 抗干扰能力分析

本节分析了 ROSMC 模型对时延、丢包、重传、抖动等因素的抗干扰能力. 首先,给出 $\boldsymbol{M}$ 被成功传输的概率计算过程. 由于 θ 为伪随机值,则 $I_1,I_2,\cdots,I_G$ 中内包数量 $A_1,A_2,\cdots,A_G$ 也为随机值. 假设 $A_1,A_2,\cdots,A_G$ 满足独立同分布,其数学期望和方差分别为 $E(A_{2j-1})=E(A_{2j})=\mu$,其中 $j=1,2,\cdots,\lfloor G/2\rfloor$. 令 $B_j=\dfrac{A_{2j-1}-A_{2j}}{2}$,则 $E(B_j)=0$,$V(B_j)\leqslant\dfrac{\sigma^2}{2}$,$A_{2j-1}$ 与 A_{2j} 均满足独立同分布,因此 B_j 也满足独立同分布.

此处仅考虑 $m_i=1$ 时的情况($m_i=0$ 的情况可类推). 在秘密消息编码器中将 m_i 嵌入流 f 之前,m_i 扩频后所对应的时隙为 $I_{h+2k-1}\sim I_{h+2k}$($h=2ir$,$k=1,2,\cdots,r$),则

$$\bar{B}=\frac{1}{r}\sum_{k=1}^{r}\frac{A_{h+2k-1}-A_{h+2k}}{2} \tag{5}$$

且 $E(\bar{B})=0$,$V(\bar{B})\leqslant\dfrac{\sigma^2}{2r}$,$\bar{B}$ 满足独立同分布.

秘密消息编码器将 m_i 嵌入流 f 后,I_{h+2k-1} 和 I_{h+2k} 内包数量分别变为 $A'_{h+2k-1}=A_{h+2k-1}+A_{h+2k-2}$ 和 A'_{h+2k},其数学期望和方差分别为

$$\left.\begin{aligned}&E(A'_{h+2k-1})=E(A_{h+2k-1}+A_{h+2k-2})=2\mu\\&E(A'_{h+2k})=0\\&V(A'_{h+2k-1})=V(A_{h+2k-1}+A_{h+2k-2})=2\sigma^2\\&V(A'_{h+2k})=0\end{aligned}\right\} \tag{6}$$

用 $\bar{B}'$ 表示发生改变的 $\bar{B}$,则 $E(\bar{B}')=\mu$,$V(\bar{B}')\leqslant\dfrac{\sigma^2}{2r}$.

1.3.1 理想网络环境

依据中心极限定理[14],$\boldsymbol{M}$ 被成功传输的概率为

$$P(\bar{B}'>0)=P\left(\frac{\bar{B}'-E(\bar{B}')}{\sqrt{V(\bar{B}')}}>\frac{-E(\bar{B}')}{\sqrt{V(\bar{B}')}}\right)\approx$$

$$1-\Phi\left(\frac{-E(\bar{B}')}{\sqrt{V(\bar{B}')}}\right)\geqslant 1-\Phi\left(\frac{-\mu\sqrt{2r}}{\sigma}\right) \tag{7}$$

式中,$\Phi(\eta)=\int_{-\infty}^{\eta}\frac{1}{\sqrt{2\pi}}e^{-\frac{u^2}{2}}du$.

1.3.2　实际网络环境

携带 $\boldsymbol{M}$ 的流 f 在实际网络传输中会受到时延、丢包等因素干扰,从而导致(I_{h+2k-1},I_{h+2k})内包数量发生改变. 设 I_{h+2k-1}受干扰后包数量的减少量为 Z_{h+2k-1}(>0),I_{h+2k}受干扰后包数量的增加量为 W_{h+2k}(>0),则受干扰后(I_{h+2k-1},I_{h+2k})包数量分别为 $A^{\mathrm{p}}_{h+2k-1}=A'_{h+2k-1}-Z_{h+2k-1}$,$A^{\mathrm{p}}_{h+2k}=A'_{h+2k}+W_{h+2k}$. 假设 Z_{h+2k-1},W_{h+2k}都满足独立同分布,其期望和方差分别为 $E(Z_{h+2k-1})=\mu_{\mathrm{Z}}$,$V(Z_{h+2k-1})=\sigma^2_{\mathrm{Z}}$,$E(W_{h+2k})=\mu_{\mathrm{W}}$,$V(W_{h+2k})=\sigma^2_{\mathrm{W}}$. 因此,$A^{\mathrm{p}}_{h+2k-1}$和 A^{p}_{h+2k}的数学期望及方差分别为

$$\left.\begin{aligned}E(A^{\mathrm{p}}_{h+2k-1})&=E(A'_{h+2k-1}-Z_{h+2k-1})=2\mu-\mu_{\mathrm{Z}}\\E(A^{\mathrm{p}}_{h+2k-1})&=E(A'_{h+2k}+W_{h+2k})=\mu_{\mathrm{W}}\\V(A^{\mathrm{p}}_{h+2k-1})&=V(A'_{h+2k-1}-Z_{h+2k-1})=2\sigma^2+\sigma^2_{\mathrm{Z}}\\V(A^{\mathrm{p}}_{h+2k-1})&=V(A'_{h+2k}+W_{h+2k})=\sigma^2_{\mathrm{W}}\end{aligned}\right\} \tag{8}$$

用 $\bar{B}^{\mathrm{p}}$ 表示发生改变的 $\bar{B}'$,则依据中心极限定理,在受干扰时 $\boldsymbol{M}$ 被成功传输的概率为

$$P[\bar{B}^{\mathrm{p}}>0]=P\left[\frac{\bar{B}^{\mathrm{p}}-E(\bar{B}^{\mathrm{p}})}{\sqrt{V(\bar{B}^{\mathrm{p}})}}>\frac{-E(\bar{B}^{\mathrm{p}})}{\sqrt{V(\bar{B}^{\mathrm{p}})}}\right]\approx 1-\Phi\left(\frac{-E(\bar{B}^{\mathrm{p}})}{\sqrt{V(\bar{B}^{\mathrm{p}})}}\right)\geqslant 1-\Phi\left(\frac{(\mu_{\mathrm{Z}}+\mu_{\mathrm{W}}-2\mu)\sqrt{r}}{\sqrt{2\sigma^2+\sigma^2_{\mathrm{Z}}+\sigma^2_{\mathrm{W}}}}\right) \tag{9}$$

式中,$E(\bar{B}^{\mathrm{p}})=\mu-\frac{\mu_{\mathrm{Z}}+\mu_{\mathrm{W}}}{2}$;$V(\bar{B}^{\mathrm{p}})=\frac{2\sigma^2+\sigma^2_{\mathrm{Z}}+\sigma^2_{\mathrm{W}}}{4r}$.

比较式(8)和式(9)可知,$P[\bar{B}^{\mathrm{p}}]>0$ 的下限值明显减小,说明受干扰情况下 $\boldsymbol{M}$ 被成功传输的概率下降. 但只要 $\mu_{\mathrm{Z}}+\mu_{\mathrm{W}}<2\mu$,式(9)中的 $\Phi(\cdot)$会随着 r 的增大而减小,$P[\bar{B}^{\mathrm{p}}]>0$ 便会增大. 由此可知,可通过增大 r 来提高 ROSMC 模型对网络干扰因素的抵抗能力.

2　实验结果与分析

2.1　实验环境与参数选择

在 Linux 系统下分别实现了秘密消息编码器与解码器,对数据包流的相关操作通过 Netfilter Iptables 软件[15]来完成,并在真实网络环境中进行测试,拓扑结构如图 4 所示. 图中,JLH,CNV 分别表示地理位置不同校区. 解码器作为应用程序运行于接收端上. 编码器的软硬件配置如下:CPU 为 Intel Xeon (R) E5606 2.13 GHz,内存为 8 GB,网卡为 NetXtreme Ⅱ BCM5709,操作系统为 Fedora 14. 接收端的软硬件配置如下:CPU 为 Intel G640 2.8 GHz,内存为 4 GB,网卡为 RTL8168FPCI-E,操作系统为 Ubuntu 12.05.

图 4　实验网络环境拓扑结构

秘密消息编码器与解码器需事先共享私密参数(θ,T,r,U). 偏移量 θ 不但会影响流 f 中的隐藏信息数量,还能保证收发端同步. 根据实验结果,设置 $\theta=50$ ms. T 选取过大,会导致 ROSMC 模型对流 f 的调整幅度较大,难以保证隐蔽性;T 过小又易受网络干扰因素影响,缺乏健壮性. 增大 r 虽然可提高 ROSMC 模型的鲁棒性,但也会使模型消耗更多的时隙与数据包,降低流 f 隐藏信息的能力. 因此,T 和 r 的设置需依据不同网络环境特性来确定. 图 5 给出了本文实验环境下传输相同 $\boldsymbol{M}$ 时 r,T 与检测正确率 R_{a} 的关系. 由图可知,R_{a} 随着 r,T

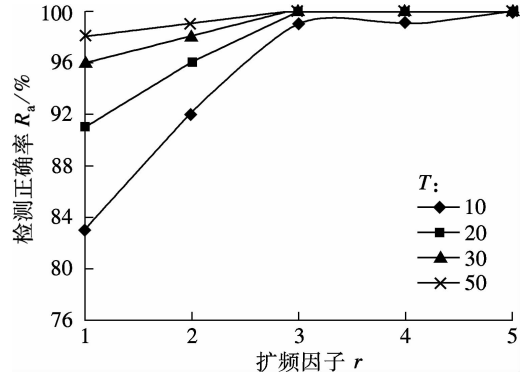


图 5　r,T 与检测正确率的关系

的增加而增大. 因此,此处设置 $r=3$,$T=20$.

2.2 结果分析

2.2.1 自同步性

ROSMC 模型通过 θ 提供同步信号. 理想条件下,θ 能准确指示载体流中 $\boldsymbol{M}$ 的起始位置;但在实际情形中,θ 会受收发主机时钟频率差异、网络延迟、抖动等因素干扰而发生改变,可能导致收发端同步失效. 然而,ROSMC 模型是在整个流 f 持续时间的若干时隙内进行 IPD 调整的,对包间时延抖动的敏感性较弱,从整体上能有效平衡延迟、抖动等干扰因素对 θ 的影响. 此外,发送端使用 MC 编码的方式来调整流 f,使其携带的 $\boldsymbol{M}$ 具有自同步性,可有效保证接收端确认 $\boldsymbol{M}$ 在 f 中的起始位置. 图 6 给出了 ROSMC 模型($\theta=50$ ms,$T=20$,$r=3$)在传输 $\boldsymbol{M}$($|\boldsymbol{M}|=24$ bit)时不同 θ 值下接收端恢复 $\boldsymbol{M}$ 的情况. 由图可知,仅在 $\theta=50$ ms 附近时,R_a 较高,对 $\boldsymbol{M}$ 的恢复效果较佳.

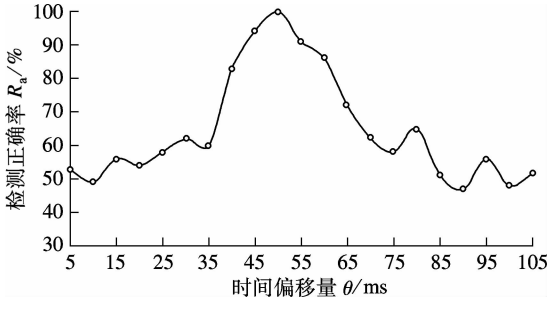


图 6 θ 与检测正确率的关系

2.2.2 鲁棒性对比

鲁棒性是指流 f 在经过网络传输后所携带秘密消息的损失程度,本质上代表了隐蔽信道的抗干扰能力. 为验证各网络隐蔽时间信道的鲁棒性,在 07:00—08:00,15:00—16:00,21:00—22:00 时段内,将不同大小的二进制文件分别用 CTCFC 算法[7]、RCTC 算法[8]、RNCC 算法[9] 及 ROSMC 算法进行收发测试,载体流为 TCP 流,结果如图 7 所示. 由图可知,在上午 07:00—08:00 时段内,校园网流量较小,4 种算法的检测错误率(即 $1-R_a$)均较低,都能较好地传输测试文件. 但在下午和晚上,校园网用户(如教师办公、学生上网)增多,网络流量增大,对隐蔽时间信道的干扰增加,RNCC 算法与 CTCFC 算法的检测错误率增大幅度远大于 RCTC 算法与 ROSMC 算法. 这是由于 RCTC 算法采用出错重传机制,ROSMC 算法采用扩频机制,它们都能较好地应对网络噪音流量的干扰. 在大流量下,虽然 RCTC 算法与 ROSMC 算法的检测错误率有所上升,但基本上都不超过 10%,其最大值较 RNCC 算法和 CTCFC 算法下降约 85%. 由此可见,RCTC 算法与 ROSMC 算法的鲁棒性更好.

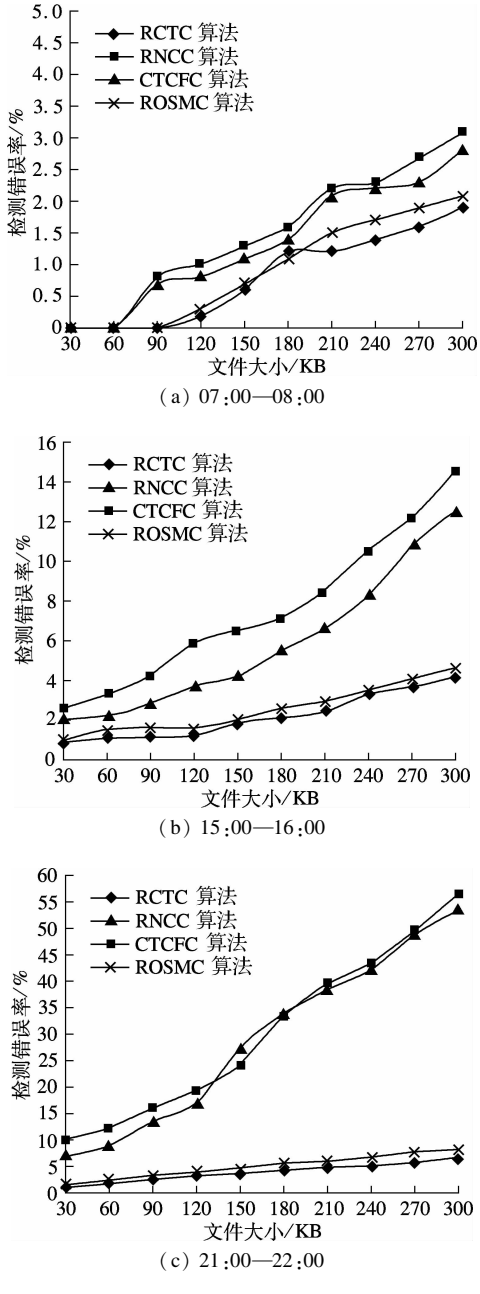


图 7 不同时段内 TCP 流量下 4 种算法的鲁棒性

图 8 展示了晚上网络流量较大时 4 种算法使用 UDP 流传输的测试结果. 由图可知,RCTC 算法的检测错误率为 100%,这是因为该算法只适用于 TCP 流. RNCC 算法的检测错误率超过 99%,原因是网络流量大且 UDP 为不可靠协议,易造成该算法中携带同步信息的数据包丢失,从而导致接收端难以准确恢复出秘密消息. 由图 7 和图 8 可知,ROSMC 算法在较大网络流量下均能较好地适用于 TCP 流和 UDP 流.

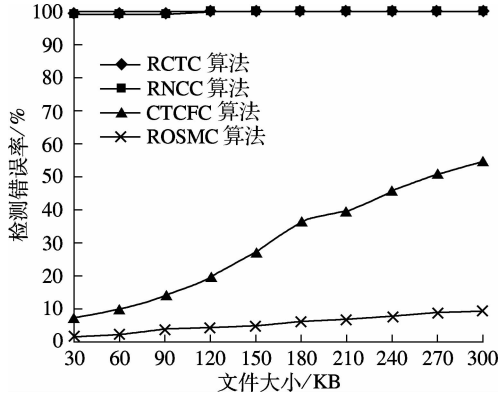


图 8　UDP 流下 4 种算法的鲁棒性

2.2.3　隐蔽性对比

隐蔽性是指载体流对携带秘密消息的暴露程度,主要用于衡量网络隐蔽时间信道应对攻击者发现能力的强弱. 隐蔽性评测主要通过 K-S(Kolmogorov-Smirnov)测试实验进行. 此处,采用双样本 K-S 测试方法对 4 种算法的隐蔽性进行测试对比,主要是通过评估 2 组样本数据是否符合相同经验累积分布函数来判断其差异.

令 $H(x)$ 为携带秘密消息的流 f 内 IPD 服从的经验累积分布函数. $G(x)$ 为正常合法网络流内 IPD 服从的经验累积分布函数,则检验统计量定义为 $d=\max_x|H(x)-G(x)|$. d 值越小,则说明经该算法处理的载体流与正常合法网络流的时间分布特征越接近,被攻击者辨别出的难度越大,由此可认为该算法的隐蔽性较好. 在不同网络流量下,分别对 4 种算法进行双样本 K-S 测试,结果见图 9. 由图可知,4 种算法的 d 值随网络流量的增大而减小,隐蔽性均呈增强趋势. 这主要是因为网络流量变大导致网络设备处理数据包时间增加,正常网络流 IPD 也增大,使得正常网络流 IPD 与含秘密消息流 f 在时间分布特征上差异减小. 此外,ROSMC 算法的隐蔽性整体上稍优于其他 3 种算法. 这是因为后者为提高鲁棒性,对载体流 IPD 的调整幅度较大,与正常网络流 IPD 显著不同,尤其是当网络流量较小时更为明显;而前者仅在载体流局部微弱调整 IPD,使其与正常流时间特征相对更为接近,因而呈现出更好的隐蔽性.

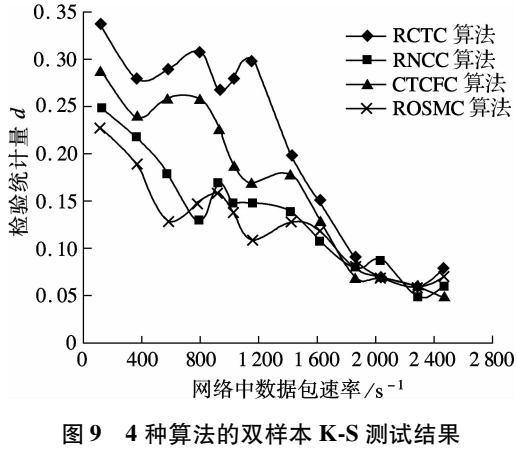


图 9　4 种算法的双样本 K-S 测试结果

3　结语

网络隐蔽时间通信本质是通过主动改变载体流时间特征来嵌入并传送数据信息的,要求能应对网络传输中各种因素干扰,保持收发端良好同步,具备较好的隐蔽性. 本文提出了一种基于 MC 的自同步网络隐蔽时间通信模型. 首先,将载体流持续时间划分为若干相同长度时隙;其次,收发端分别通过调整和判断相邻时隙内包数量来模拟 MC 编解码操作;然后,在扩频机制和时间偏移量辅助下,完成秘密信息的嵌入、传输及恢复过程. 实验结果表明,该方法的鲁棒性、同步性均优于传统 IPD 时间隐蔽信道,且在网络负载较大时仍能保持较好的隐蔽性. 下一步将就信道容量的提升、无线网络中的适应性、与相关网络安全措施的结合性等方面展开研究工作.

参考文献(References)

[1] 冯登国,张敏,李昊. 大数据安全与隐私保护[J]. 计算机学报,2014,37(1):246－258.
Feng Dengguo, Zhang Min, Li Hao. Big data security and privacy protection[J]. *Chinese Journal of Computers*, 2014, **37**(1): 246－258. (in Chinese)

[2] 冯登国,张敏,张妍,等. 云计算安全研究[J]. 软件学报,2011,22(1):71－83.
Feng Dengguo, Zhang Min, Zhang Yan, et al. Study on cloud computing security[J]. *Journal of Software*, 2011, **22**(1): 71－83. (in Chinese)

[3] 谭庆丰,刘培朋,时金桥,等. UGC³:一种抵御审查的隐蔽通信方法[J]. 通信学报,2012,33(8):155－161.
Tan Qingfeng, Liu Peipeng, Shi Jinqiao, et al. UGC³: a covert communication method defense against censorship[J]. *Journal on Communications*, 2012, **33**(8): 155－161. (in Chinese)

[4] Rios R, Onieva J A, Lopez J. Covert communications through network configuration messages[J]. *Computers & Security*, 2013, **39**: 34－46.

[5] 章思宇,邹福泰,王鲁华,等. 基于 DNS 的隐蔽通道流量检测[J]. 通信学报,2013,34(5):143－151.
Zhang Siyu, Zhou Futai, Wang Luhua, et al. Detecting DNS-based covert channel on live traffic[J]. *Journal on Communications*, 2013, **34**(5): 143－151. (in Chinese)

[6] Cabuk S, Brodley C E, Shields C, et al. IP covert timing channels: design and detection[C]//*Proceedings of the 11th ACM Conference on Computer and Communications Security*. New York: ACM, 2004: 178 - 187.

[7] Archibald R, Ghosal D. A covert timing channel based on fountain codes[C]//2012 *IEEE International Conference on Trust, Security and Privacy in Computing and Communications*. Liverpool, UK, 2012: 970 - 977.

[8] 钱玉文,赵邦信,孔建寿,等. 一种基于 Web 的可靠网络隐蔽时间信道的研究[J]. 计算机研究与发展, 2011,48(3):423 - 431.
Qian Yuwen, Zhao Bangxin, Kong Jianshou, et al. Robust covert timing channel based on Web[J]. *Journal of Computer Research and Development*, 2011, **48**(3): 423 - 431. (in Chinese)

[9] 牛小鹏,李清宝,王炜. 一种基于扩频编码的可靠网络隐蔽信道设计方法[J]. 电子与信息学报,2013,35(4):1012 - 1016.
Niu Xiaopeng, Li Qingbao, Wang Wei. A robust network covert channel algorithm based on spread coding [J]. *Journal of Electronics & Information Technology*, 2013, **35**(4): 1012 - 1016. (in Chinese)

[10] Zhang Z, Guo Z, Yang Y. Bounded-reorder packet scheduling in optical cut-through switch [C]//2013 *IEEE INFOCOM*. Turin, Italy, 2013: 701 - 709.

[11] Narasiodeyar R M, Jayasumana A P. Improvement in packet-reordering with limited re-sequencing buffers: an analysis [C]//2013 *IEEE Conference on Local Computer Networks*. Sydney, Australia, 2013: 416 - 424.

[12] Liu Y, Ghosal D, Armknecht F, et al. Robust and undetectable steganographic timing channels for i. i. d. traffic[C]//*Information Hiding Conference*. Calgary, Canada, 2010: 193 - 207.

[13] Giustiniano D, Lenders V, Schmitt J B, et al. Detection of reactive jamming in DSSS-based wireless networks[C]//*Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks*. Budapest, Hungary, 2013: 43 - 48.

[14] 谢安,李冬红. 概率论与数理统计[M]. 北京:清华大学出版社,2012:122 - 125.

[15] Netfilter Core Team. The netfilter. org "iptables" project [EB/OL]. (2013-11-22)[2014-06-15]. http://www. netfilter. org. /projects/iptables/index. html.