doi:10.3969/j.issn.1000-436X.2012.10.014

分布式时间戳同步技术的改进

林楷[1]，贾春福[1]，石乐义[2]
（1. 南开大学 信息技术科学学院，天津 300071；2. 中国石油大学（华东） 计算机与通信工程学院，山东 青岛 266555）

摘 要：分布式时间戳同步（DTS, distributed timestamp synchronization）技术能够较好地满足端信息跳变的同步需求，但仍存在一定程度的同步失败。对 DTS 技术进行了改进（IDTS, improved DTS）：额外开启一个前置和一个后置端信息用于接收同步失败的数据分组。首先构建了端信息跳变系统的服务模型并给出了 IDTS 技术的通信协议，在此基础上分析说明了 IDTS 技术的有效性和安全性，最后通过实验验证了 IDTS 技术的实践价值。
关键词：网络防护；端信息跳变；同步技术；分布式时间戳同步技术
中图分类号：TP393.08 **文献标识码**：A **文章编号**：1000-436X(2012)10-0110-07

Improvement of distributed timestamp synchronization

LIN Kai[1], JIA Chun-fu[1], SHI Le-yi[2]
(1. College of Information Technical Science, Nankai University, Tianjin 300071, China;
2. College of Computer and Communication Engineering, China University of Petroleum, Qingdao 266555, China)

Abstract: Distributed timestamp synchronization (DTS) could satisfy the requirement of synchronization for end hopping, but still had some degree failure of synchronization. An improved scheme of DTS (say IDTS) was proposed by providing two additional endpoints to receive the unsynchronized packets. It established models and definitions for end hopping system at first, then provided protocol for IDTS，analyzed the effectiveness and security of IDTS. Finally, the empirical results show the practical value of IDTS.
Key words: network defense; end hopping; synchronization technology; distributed timestamp synchronization

1 引言

传统的网络防护技术难以满足人们日益增长的安全需求[1,2]，如防火墙和入侵检测等技术。于是，人们越来越关注更为主动、积极的防护技术，如蜜罐[3,4]和端信息跳变[5]等技术。在端信息跳变技术中，跳变系统不断改变网络服务的网络信息（如 IP 地址、端口、协议和服务程序等），使攻击者无法确定攻击目标，从而达到保护网络服务的目的。同步技术是端信息跳变技术的核心技术之一[6]，决定着跳变系统的抗攻击性能和受保护目标系统的服务性能。

在无线跳频通信中[7,8]，将时间划分为等长的时间片并为每个时间片分配一个固定的频率，形成一张时间与频率之间的映射表。其中，由于无线通信的传输延迟很小（接近于零），很少出现同步失败的情况。然而，在存在大量网络拥塞和延迟的 Internet 网络中，基于严格时间分片的高耦合度同步

收稿日期：2011-12-26；**修回日期**：2012-09-11
基金项目：国家自然科学基金资助项目（60973141, 61272423）；国家重点基础研究发展计划（“973”计划）基金资助项目（2013CB834204）；高等学校博士学科点专项科研基金资助项目（20100031110030）
Foundation Items: The National Natural Science Foundation of China (60973141, 61272423) ; The National Key Basic Research Program of China (973 Program) (2013CB834204); The Specialized Research Fund for the Doctoral Program of Higher Education of China (20100031110030)

技术难以适用于端信息跳变技术。

在无线传感器网络通信[9,10]中，各个传感器节点往往需要基于时间逻辑顺序的协同工作，节点需要周期性地进行时钟同步以校正节点的时间。传感器网络一般处于小型局域网环境中，节点间的传输延迟较小且平稳，可以利用 NTP 或者 PTP 时钟同步协议达到高精度的时钟同步。然而，在 Internet 网络中，NTP/PTP 时钟同步协议难以达到局域网环境的同步精度，因此简单的基于 NTP/PTP 时钟同步协议的同步技术也难以适用于端信息跳变技术。

在 Gal 等[11]提出的跳端口技术中，Ack-based 同步技术利用已经成功发送的数据分组的数目和共享的私钥计算得出本次通信所采用的端口。这种同步技术不需要时间上的严格耦合，而依赖于已成功传输的数据分组的数目，解决了网络拥塞和延迟等问题。但其仍存在许多不足之处，主要包括：对“一对多”的通信方式支持不足，需要维护大量的同步消息；跳变的参数仅包含端口，如果攻击者能够过载通信一方的上级路由节点，依然能够成功实现攻击；仅适用于类似于 UDP 协议的无连接数据通信模式；需要修改上下层网络协议，兼容性难以保证。

在端信息跳变技术中，石乐义等[5]提出了时间戳同步技术，利用时间戳计算出端信息，并提供一个统一的时间戳服务器负责分发时间戳。然而，这种集中式的时间戳分发机制存在很大安全风险，一旦攻击者成功入侵时间戳服务器，攻击者将会很容易瓦解跳变系统。

为此，文献[12]提出了分布式时间戳同步（DTS, distributed timestamp synchronization）技术，在时间戳同步技术的基础之上，将时间戳分发的过程分布式地部署在 Internet 网络上。这不但能够解决网络拥塞和延迟等问题，也使得跳变技术的安全性能摆脱了时间戳服务器的约束。然而，即使在没有受到网络攻击的情况下，DTS 仍存在一定程度的同步失败率 $P_{\text{failure}} = \max\{\Phi, \Delta\}/\delta$，其中，$\delta$为跳变时隙，$\Phi$为最大时间漂移，$\Delta$为最大传输延迟[12]。$\Phi$和$\Delta$受计算机系统和所处网络环境的影响，因此降低同步失败率的方法只有增加δ，但这也意味着安全性能的下降。

本文分析了 DTS 的缺陷并对其做出了改进，提出了 DTS 的改进方案 IDTS，分析了 IDTS 的有效性和安全性，并通过真实的网络环境下的攻防实验验证了分析结论的正确性。

2　跳变系统的建模与 IDTS 方案

2.1　几个重要的概念

最大传输延迟Δ：数据分组的最长传输时间。无论 2 个网络节点相隔多远（路由跳数），数据分组的传输延迟应小于或等于最大传输延迟，否则认为传输失败。

时间戳：计算机系统的本地时间；实际时间：指现实世界的时间，而不是钟表仪器上显示的刻度；最大时间漂移Φ：节点的时间戳与实际时间的最大偏差。计算机系统的时间戳 t_{loc} 和实际时间 t_{real} 之间总存在一定的漂移，且满足 $0 \leqslant |t_{\text{loc}} - t_{\text{real}}| \leqslant \Phi$。$t_{\text{loc}}(t)$是指计算机在任意实际时间 $t_{\text{real}}=t$ 的时间戳。

端信息ε：网络服务的信息或参数，一般包括 IP 地址和端口，即$\varepsilon = (IP, Port)$。数据分组有目的端信息$\varepsilon_{\text{dest}}$和源端信息$\varepsilon_{\text{source}}$。$\varepsilon(t_{\text{loc}}(t))$是指在任意实际时间 t 根据计算机的时间戳计算而得到的端信息。

端信息的状态空间E：跳变系统中所有可用端信息的集合。端信息的状态空间与跳变系统所配置的软硬件有关，若跳变系统有 IP 地址集合 $\boldsymbol{A} = \{IP_1, \cdots, IP_n\}$和端口集合$\boldsymbol{\Psi} = \{Port_1, \cdots, Port_\psi\}$，则端信息的状态空间$\boldsymbol{E}=\boldsymbol{A}\times\boldsymbol{\Psi} = \{(IP_i, Port_j) \mid 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant \psi\}$。

跳变事件：指跳变系统更新端信息。

跳变时隙δ：2 个相邻跳变事件之间的时间且满足$\delta > 2\Phi + \Delta$，因为如果$\delta \leqslant 2\Phi + \Delta$，那么数据分组的传输时间总会横跨一个或多个跳变时隙，本文不考虑跨多个时隙的情况。

最大服务性能 I：在一个跳变时隙之内，单个端信息能够响应的最大数据流量（文中为了简化模型，将服务性能简单地等价于数据流量）。如果某个端信息接收到的数据流量 $R \leqslant I$，则所有的服务请求流量都能够得到端信息的服务响应，服务率为 1（100%）；如果 $R > I$，那么只有数据流量 R 的一部分能够得到端信息的服务响应，服务率为 I/R。

服务率μ：客户端数据分组得到响应的概率。对于本文中的网络服务，有 2 点假设：第一，服务不存在崩溃型拒绝服务漏洞（指只要少量的攻击流量就会造成服务崩溃）；第二，服务是无损的（只要服务请求流量不超过 I，则μ为 1）。

最大攻击强度 C：在一个跳变时隙之内，攻击者所能产生的最大攻击流量。

攻击策略 S：攻击者所采取的攻击策略。$S(C)=\{\{c_1, c_2,\cdots, c_{|E|}\}|\sum c_i \leqslant C, i=1,2,\cdots,|E|\}$指的是在端信息状态空间为 E 和最大攻击强度为 C 条件下的攻击策略。

期望攻击强度 U_S：在攻击策略 $S(C)$下，单个端信息所受到攻击流量的期望值。

2.2 改进方案 IDTS

DTS 同步技术是在时间戳同步技术基础之上提出的，利用分布式的时间戳服务器分发时间戳，解决在集中时间戳服务模式中的安全瓶颈。在 Internet 上部署多个独立的时间戳服务器，其中每个服务器能够独立提供时间戳分发服务。跳变系统和客户端分别维护自己的时间戳服务器列表，并周期性地从列表中选择一个或多个服务器来更新本地时间戳。

DTS 将时间分割成等长的时间片（即跳变时隙）并编号，所有属于同一时间片的时间戳都可以利用单向散列函数(PRF)映射到同一个端信息，$\varepsilon(t_{loc})=PRF(slot(t_{loc}), key)$，其中，$slot(t_{loc})=\lfloor t_{loc}/\delta \rfloor$，$key$ 为跳变系统和客户端共享的密钥。在任意实际时间 t，客户端根据其时间戳 $t_c(t)$计算出数据分组的目的端信息$\varepsilon(t_c(t))$，跳变系统也根据其时间戳 $t_s(t)$计算出服务端信息$\varepsilon(t_s(t))$。当数据分组到达跳变系统时，只要满足$\varepsilon(t_c(t))=\varepsilon(t_s(t))$, 则认为同步成功。然而，由于存在时间漂移和传输延迟，而且跳变系统和客户端分别与不同的时间戳服务器进行同步，因此，一般情况 $t_c(t)\neq t_s(t)$。为了使同一时间片内的不同时间戳能够映射到同一端信息，即对于$\forall t_c(t)$, $t_s(t)\in[m\delta, (m+1)\delta)) \rightarrow \varepsilon(t_c(t))=\varepsilon(t_s(t))=\varepsilon(m\delta)$，对时间戳 t_{loc} 进行时隙处理 $slot(t_{loc})$，将 t_{loc} 映射为对应的跳变时隙编号[12]。

然而，DTS 存在一定程度的同步失败。例如，令 t_0 和 t_1 分别为客户端发出数据分组和跳变系统接收数据分组的实际时间，如果 $t_c(t_0)\in[m\delta, (m+1)\delta))$，而 $t_s(t_1)\in[(m-1)\delta, m\delta))$或 $t_s(t_1)\in[(m+1)\delta, (m+2)\delta))$，则 $slot(t_c(t_0))\neq slot(t_s(t_1))$，即$\varepsilon(t_c(t_0))\neq\varepsilon(t_s(t_1))$，所以同步失败。

为此，本文提出了 DTS 技术的改进方案（IDTS）：依据跳变时隙的时间顺序，在原基础之上，额外开启一个前置端信息ε_{prev}和一个后置端信息ε_{next}。例如，跳变系统的当前时间戳对应着跳变时隙 i，则额外开启第 $i-1$ 和 $i+1$ 个跳变时隙所对应的端信息。令 $E^*=\{\varepsilon_{prev}, \varepsilon_{curr}, \varepsilon_{next}\}$为 IDTS 中所开启的端信息集合。

在 IDTS 中，跳变系统与客户端的通信协议如图 1 所示，分别包括跳变系统和客户端。只要密钥 key 是安全的，则基于 IDTS 的通信就是安全的。跳变系统和所有的客户端都共享着一个同步密钥 key。

```
On  sys_init(ts(t)) :
s = slot(ts(t));
εprev = PRF(s-1, key);   alloc(εprev, I);
εnow = PRF(s, key);      alloc(εnow, I);
εnext = PRF(s+1, key);   alloc(εnext, I);
On  recv_event(pkt) :                    // pkt 是数据分组
if(validate(pkt)) then service (pkt);
On  hopping(ts(t))  where  ts(t) ∈ {0, δ, 2δ, …} :
alloc (εprev, 0);                        // 关闭
εprev = εnow; εnow = εnext;
s = slot(ts(t));
εnext = PRF (s+1, key);   alloc (εnext, I);
On  send_event (pkt) :
pkt.εsource = εnow;              send (pkt)
```

(a) 跳变系统

```
On  sys_init(tc(t)) :
random(εloc);          alloc(εloc, I*);   // 接收数据
s = slot(tc(t));       εnow = PRF(s, key);
On  recv_event(pkt) :
if(validate(pkt)) then post(pkt);
On  hopping(tc(t))  where  tc(t) ∈ {0, δ, 2δ, …}:
s = slot(tc(t));       εnow = PRF(s, key);
On  send_event(pkt) :
pkt.εdest = εnow;              send(pkt)
```

(b) 客户端

图 1 IDTS 同步协议

3 IDTS 的分析

3.1 服务性能分析

令 $t_s(t)$为跳变系统的时间戳，$t_c(t)$为客户端的时间戳，则 $0\leqslant|t_s(t)-t|\leqslant\Phi$，$0\leqslant|t_c(t)-t|\leqslant\Phi$。因此，$t_s(t)$和 $t_c(t)$可能存在以下 2 种情况。

情况 1 $t_s(t)<t_c(t)$。即存在θ，使 $t_s(t)=t_c(t)-\theta$，$0<\theta<2\Phi$。数据分组传输过程如图 2 所示，其中 2 条带箭头的粗线分别代表客户端和跳变系统的本地时间轴，带编号的有向线指出了在一个跳变时隙δ之内数据分组发收状态，m 为任意大于 0 的整数。

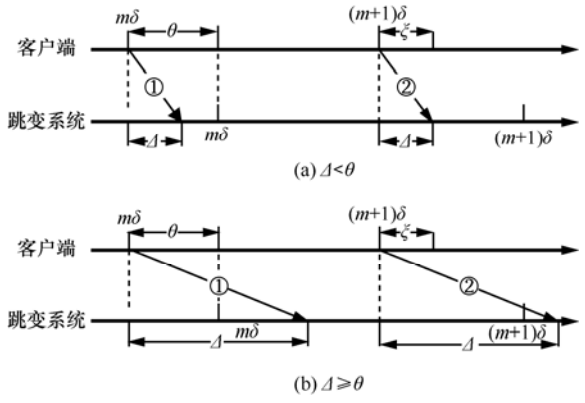


图 2　$t_s(t)=t_c(t)-\theta$时，数据分组传输状态

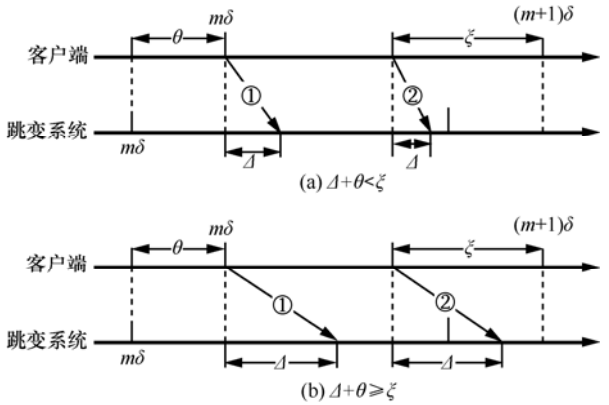


图 3　$t_s(t)=t_c(t)+\theta$ 时，数据分组传输状态

在跳变时隙之初（如图 2 中①所示），令 t_0 为数据分组发出的实际时间且 $t_c(t_0)=m\delta$，则目的端信息为$\varepsilon_{dest}(t_c(t_0))=\varepsilon(m\delta)$。在实际时间 $t_0+\Delta$，数据分组达到跳变系统，此时 $t_s(t_0+\Delta)=t_c(t_0+\Delta)-\theta=m\delta+\Delta-\theta$。由$\delta>2\Phi+\Delta$，可得$\Delta-\theta\in(\Delta-2\Phi,\Delta)\subset(-\delta,\delta)$，若$\Delta-\theta\in(-\delta,0)$（如图 2(a)中①所示），则 $t_s(t_0+\Delta)\in[(m-1)\delta,m\delta)$，于是 $E^*=\{\varepsilon((m-2)\delta),\varepsilon((m-1)\delta),\varepsilon(m\delta)\}$，$\varepsilon_{dest}(t_c(t_0))\in E^*$；若$\Delta-\theta\in[0,\delta)$（如图 2(b)中①），则 $t_s(t_0+\Delta)\in[m\delta,(m+1)\delta)$，于是 $E^*=\{\varepsilon((m-1)\delta),\varepsilon(m\delta),\varepsilon((m+1)\delta)\}$，$\varepsilon_{dest}(t_c(t_0))\in E^*$。

在跳变时隙之末（如图 2 中②所示），令 t_0 为数据分组发出的实际时间且 $t_c(t_0)=(m+1)\delta-\xi$，$0<\xi<\delta-\Delta-2\Phi$，则目的端信息为$\varepsilon_{dest}(t_c(t_0))=\varepsilon_{dest}((m+1)\delta-\xi)=\varepsilon(m\delta)$。在实际时间 $t_0+\Delta$，数据分组达到跳变系统，此时 $t_s(t_0+\Delta)=t_c(t_0+\Delta)-\theta=(m+1)\delta+\Delta-\theta-\xi$。由$\delta>2\Phi+\Delta$，可得$\Delta-\theta-\xi\in(2\Delta-\delta,\Delta)\subset(-\delta,\delta)$，若$\Delta-\xi-\theta\in(-\delta,0)$（如图 2(a)中②），则 $t_s(t_0+\Delta)\in[m\delta,(m+1)\delta)$，于是 $E^*=\{\varepsilon((m-1)\delta),\varepsilon(m\delta),\varepsilon((m+1)\delta)\}$，$\varepsilon_{dest}(t_c(t_0))\in E^*$；若$\Delta-\xi-\theta\in[0,\delta)$（如图 2(b)中②所示），则 $t_s(t_0+\Delta)\in[(m+1)\delta,(m+2)\delta)$，于是 $E^*=\{\varepsilon(m\delta),\varepsilon((m+1)\delta),\varepsilon((m+2)\delta)\}$，$\varepsilon_{dest}(t_c(t_0))\in E^*$。

情况 2　$t_s(t)\geqslant t_c(t)$。存在θ，使在 t 时刻，$t_s(t)=t_c(t)+\theta$，$0<\theta<2\Phi$，传输过程如图 3 所示。

在跳变时隙之初（如图 3 中①所示），令 t_0 为客户端发出数据分组的实际时间且 $t_c(t_0)=m\delta$，则目的端信息为$\varepsilon_{dest}(t_c(t_0))=\varepsilon(m\delta)$。在实际时间 $t_0+\Delta$，数据分组达到跳变系统，此时 $t_s(t_0+\Delta)=t_c(t_0)+\theta=m\delta+\Delta+\theta$，$\Delta+\theta\in[0,\delta)$，因此 $t_s(t_0+\Delta)\in[m\delta,(m+1)\delta)$，于是 $E^*=\{\varepsilon((m-1)\delta),\varepsilon(m\delta),\varepsilon((m+1)\delta)\}$，$\varepsilon_{dest}(t_c(t_0))\in E^*$。

在跳变阶段之末（如图 3 中②所示），令 t_0 为数据分组发出的实际时间且 $t_c(t_0)=(m+1)\delta-\xi$，$0<\xi<\delta-\Delta-2\Phi$，则目的端信息为$\varepsilon_{dest}(t_c(t_0))=\varepsilon((m+1)\delta-\xi)=\varepsilon(m\delta)$。在实际时间 $t_0+\Delta$，数据分组达到跳变系统，此时 $t_s(t_0+\Delta)=t_c(t_0)+\theta=(m+1)\delta+\Delta+\theta-\xi$。由$\delta>2\Phi+\Delta$，可得$\Delta+\theta-\xi\in(2\Phi+2\Delta-\delta,\Delta+2\Phi)\subset(-\delta,\delta)$。此时，若$\Delta+\theta-\xi\in(-\delta,0)$（如图 3(a)中②所示），则 $t_s(t_0+\Delta)\in[m\delta,(m+1)\delta)$，于是 $E^*=\{\varepsilon((m-1)\delta),\varepsilon(m\delta),\varepsilon((m+1)\delta)\}$，$\varepsilon_{dest}(t_c(t_0))\in E^*$；若$\Delta+\theta-\xi\in[0,\delta)$（如图 3(b) 所示中②所示），则 $t_s(t_0+\Delta)\in[(m+1)\delta,(m+2)\delta)$，于是 $E^*=\{\varepsilon(m\delta),\varepsilon((m+1)\delta),\varepsilon((m+2)\delta)\}$，$\varepsilon_{dest}(t_c(t_0))\in E^*$。

3.2　安全性能分析

攻击者有 2 种可供选择的攻击方式：盲攻击和窃听攻击[11]。

1) 盲攻击

在盲攻击中，攻击者无法得知通信所采用的端信息，于是采用随机选取的盲攻击方式。攻击者随机选取一个或多个端信息进行攻击，即从 E 中随机地选择 l（$l>0$）个端信息作为攻击目标，则任一端信息被选中的概率 $P_g=l/|E|$，每个端信息受到的攻击流量 $c=C/l$。因此单个端信息所受到的期望攻击强度 $U_S=cP_g=C/|E|$。因此，在受到盲攻击时，服务率μ满足如下公式：

$$\mu(q,I,C)=\begin{cases}1,\ q+C/|E|\leqslant I\\ \dfrac{I}{q+C/|E|},\ q+C/|E|>I\end{cases}$$

其中，q 为客户端流量，I 为单个端信息的最大服务性能，C 为攻击者的最大攻击强度。

在盲攻击中，当 q 和 I 固定不变时，服务率由 $C/|E|$决定，而$|E|=|A||\Psi|$。

2) 窃听攻击

窃听攻击指的是攻击者利用嗅探技术捕获客户端与跳变系统的通信数据分组，并对数据分组进行分析处理，进而获取跳变系统所开启的端信息。

由于网络传输时间都为Δ，在数据分组到达跳变系统时，也同时被攻击者截获。假设攻击者需要花费时间γ来进行攻击准备（称γ为攻击者的攻击耗时）。为了简化攻击耗时与服务率之间关系的分析过程，考虑特殊情况：$t_s(t)=t_c(t)-\theta$，$\theta>0$ 且$\Delta=\theta$，如图 4 所示，客户端与跳变系统完全同步。

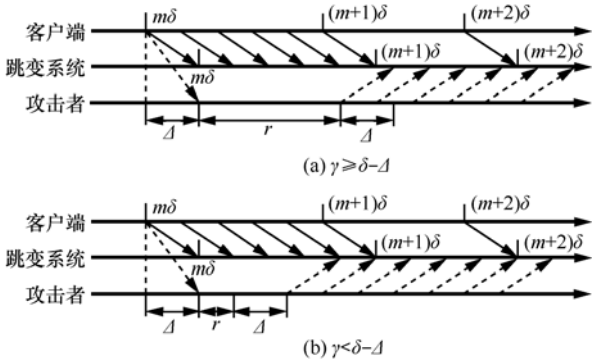


图 4 窃听攻击的时间轴状态

如图 4 中所描绘的情况，令 t_0 为客户端数据分组发出的实际时间且 $t_c(t_0)=m\delta$，则目的端信息为 $\varepsilon_{dest}(t_c(t_0))=\varepsilon(m\delta)$。攻击者在实际时间 $t_0+\Delta$截获该数据分组，经过时间γ的分析处理后发出攻击数据分组到端信息$\varepsilon(m\delta)$。最后，在实际时间 $t_0+2\Delta+\gamma$，攻击数据分组到达跳变系统。

若$\gamma\in[\delta-\Delta,\infty)$（如图 4(a)所示），假设存在正整数 n，使$\gamma\in[n\delta-\Delta,(n+1)\delta-\Delta)$，则 $t_s(t_0+2\Delta+\gamma)=t_c(t_0+2\Delta+\gamma)-\theta=m\delta+\gamma+\Delta\in[(m+n)\delta,(m+n+1)\delta)$。这意味着当攻击数据分组达到跳变系统时，客户端利用$\varepsilon((m+n)\delta)$进行通信，因此，发向$\varepsilon(m\delta)$的攻击数据分组无法影响客户端的数据分组，服务率为 1。

若$\gamma\in[0,\delta-\Delta)$（如图 4(b)所示），则 $t_s(t_0+2\Delta+\gamma)=m\delta+\gamma+\Delta\in[m\delta+\Delta,(m+1)\delta)$。如果客户端数据分组在$\forall t_s(t)\in[m\delta,m\delta+\gamma+\Delta)$内到达跳变系统，不会受到攻击数据分组的影响，服务率为$\mu_1=1$；如果客户端数据分组在$\forall t_s(t)\in[m\delta+\gamma+\Delta,(m+1)\delta)$到达跳变系统，则只有部分数据分组能够得到跳变系统的响应。假设攻击数据分组和客户端数据分组以均匀速率达到跳变系统，则期望攻击强度 U_S 为 $C(\delta-\gamma-\Delta)/\delta$，客户端流量 q_c 为 $q(\delta-\gamma-\Delta)/\delta$，服务率$\mu_2=I/(q_c+U_S)=I\delta/((\delta-\gamma-\Delta)(q+C))$。因此，综合服务率为

$$\begin{aligned}\mu&=p(\mu_1)\mu_1+p(\mu_2)\mu_2\\&=(\gamma+\Delta)/\delta+((\delta-\gamma-\Delta)/\delta)(I\delta/((\delta-\gamma-\Delta)(q+C)))\\&=(\gamma+\Delta)/\delta+I/(q+C)\\&\approx(\gamma+\Delta)/\delta\quad(\text{因为 } I/(q+C)\approx 0)\end{aligned}$$

因此，在窃听攻击中，服务率由跳变时隙和攻击耗时共同决定，攻击耗时越高，服务率越高。

4 攻防实验

在局域网环境下，分别搭建了基于 DTS 技术和 IDTS 技术的端信息跳变系统。跳变系统和客户端的系统为 Arch-Linux（内核版本 2.6.32，双核处理器：2.66GHz，内存：1GB），跳变系统的 IP 地址集合为{12.12.12.30, 12.12.12.31}，端口集合为{2 000, 2 001,…, 3 000}，最大服务性能 I 为 200kbit/s，客户端流量 q 为 30kbit/s，攻击者的最大攻击强度 C 为 1 000kbit/s，最大时间漂移Φ为 40ms，数据分组的最大传送延迟Δ为 2ms。

4.1 盲攻击实验

在盲攻击实验中，采用 3 种跳变时隙（200ms、500ms 和 1 000ms），对跳变系统进行了 3 组实验，每组实验分别包含 500, 1 000, … , 5 500 个服务请求。实验结果如图 5 所示。

由实验结果可得，IDTS 的服务率接近 1，而且不受跳变时隙长短的影响。然而，DTS 技术由于受时间漂移和传输延迟的影响，服务率不足 1，且依赖于跳变时隙的长短：跳变时隙越短，服务率越低。

4.2 窃听攻击实验

在窃听攻击实验中，对跳变系统（采用了 3 个不同跳变时隙 200ms、500ms 和 1 000ms）进行请求实验，攻击者发送攻击数据分组给所截获的端信息，客户端发送请求数据分组，分别测试了不同攻击耗时情况下的服务率，结果如图 6 所示。

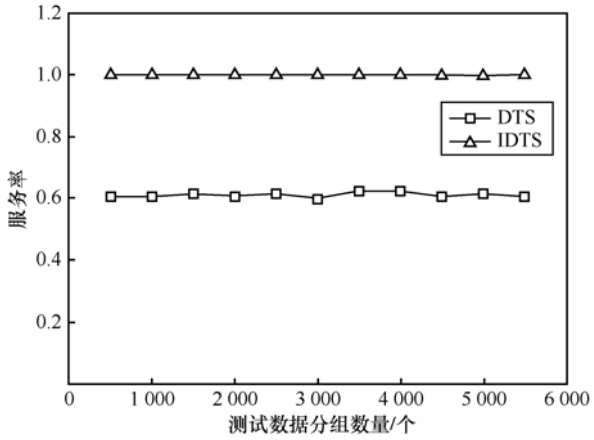
由实验结果可以得出以下结论。

1) 攻击耗时越短，跳变系统的服务率越低，攻击效果越好。

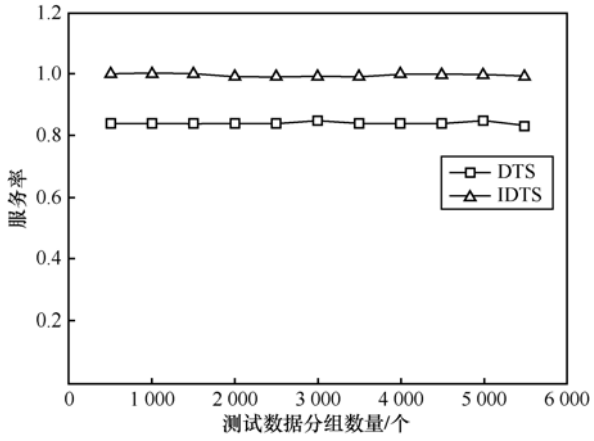
2) 当攻击耗时下降为 0ms 时，DTS 同步技术的服务率μ下降为 0，而 IDTS 同步技术仍能够保证一定程度服务率μ，其原因是因为客户端与跳

变系统之间的存在时间漂移，使得客户端数据分组发送到了前置端信息，而攻击者仅仅攻击当前端信息。

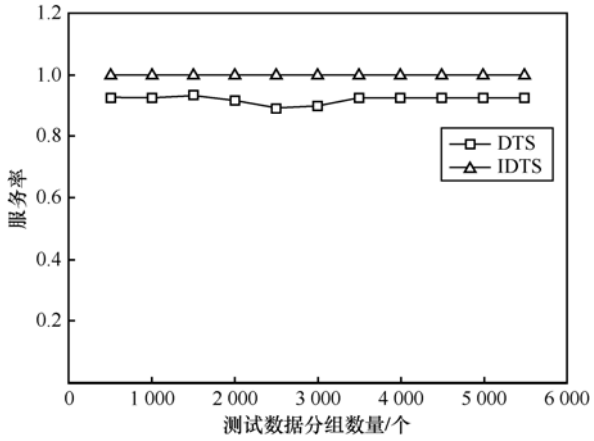
3) IDTS 技术减轻了跳变系统对跳变时隙的依赖程度。DTS 技术过度依赖于跳变时隙的长短：随着跳变时隙的变长，服务率也增高。但是，随着时隙的变长，安全性能也随之下降。



(a) 200 ms 时隙

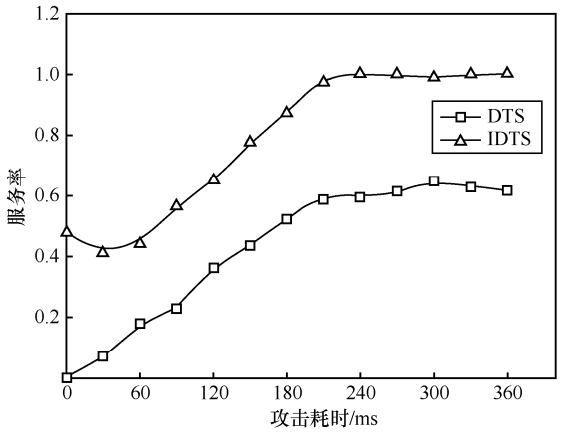


(b) 500 ms 时隙

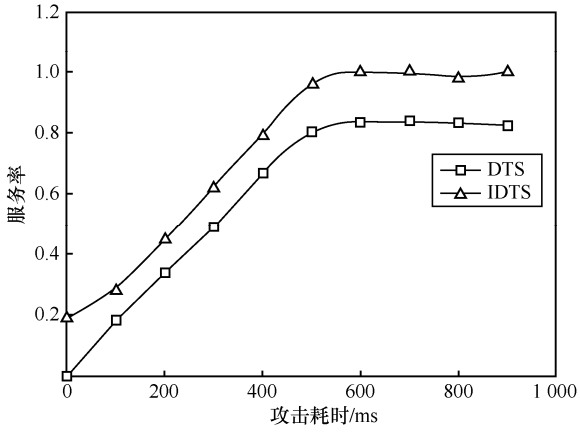


(c) 1 000 ms 时隙

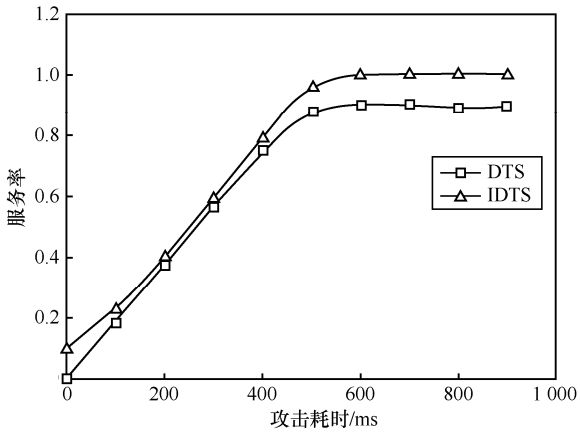
图 5　盲攻击实验



(a) 200 ms 时隙



(b) 500 ms 时隙



(c) 1 000 ms 时隙

图 6　窃听攻击实验

5　结束语

本文对端信息跳变系统进行了建模，在此基础上，分析了分布式时间戳同步技术的通信过程，并阐述了同步失败的原因。进而给出了分布式时间戳同步技术的改进方案，并对 IDTS 分别进行了服务性能和安全性能 2 个方面的分析。在服务性能分析中，分别从 $t_s(t) < t_c(t)$ 和 $t_s(t) \geqslant t_c(t)$ 2 种情况分析了

IDTS 在服务过程中的同步情况。在安全性能分析中，从攻击者的角度，分别对盲攻击和窃听攻击的攻击效果进行推理分析。结果表明 IDTS 拥有更好的服务性能和安全性能。

最后，在真实的网络环境中对比了 DTS 和 IDTS 的攻防服务过程。实验结果表明 IDTS 拥有较高的实践价值。

参考文献：

[1] 杨雅辉, 姜电波, 沈晴霓等. 基于改进的 GHSOM 的入侵检测研究[J]. 通信学报, 2011, 32(1): 121-126.
YANG Y H, JIANG D B, SHEN Q N, *et al*. Research on intrusion detection based on an improved GHSOM[J]. Journal on Communications, 2011, 32(1): 121-126.

[2] 钟婷, 刘勇, 李志军等. 基于网络处理器的 IPv4/IPv6 综合防火墙体系结构研究[J]. 通信学报, 2006, 27(2):142-146.
ZHONG T, LIU Y, LI Z J, *et al*. Research of a comprehensive IPv4/IPv6 firewall system based on network processor[J]. Journal on Communications, 2006, 27(2): 142-146.

[3] PING W, LEI W, RYAN C. Honeypot detection in advanced botnet attacks[J]. International Journal of Information and Computer Security, 2010, 4(1): 30-51.

[4] LEE K, CAVERLEE J, WEBB S. The social honeypot project: protecting online communities from spammers[A]. Proceedings of The 19th International Conference on World Wide Web[C]. Piscataway, NY, USA: IEEE, 2010. 1139-1140.

[5] 石乐义, 贾春福, 吕述望. 基于端信息跳变的主动网络防护研究[J]. 通信学报, 2008, 29(2): 106-110.
SHI L Y, JIA C F, LV S W. Research on end hopping for active network confrontation[J]. Journal on Communications, 2008, 29(2): 106-110.

[6] 石乐义, 贾春福, 吕述望. 服务跳变抗 DoS 机制的博弈理论分析[J]. 电子与信息学报, 2009, 31(1): 228-232.
SHI L Y, JIA C F, LV S W. A game theoretic analysis of service hopping mechanism for DoS defense[J]. Journal of Electronics & Information Technology, 2009, 31(1): 228-232.

[7] FUJI R, FUJIWARA Y, JIMBO M, *et al*. Sets of frequency hopping sequences: bounds and optimal constructions[J]. IEEE Transactions on Information Theory, 2009, 55(7): 3297-3304.

[8] JIAN Z, XIN Y W, XIAO G N. Performance analysis of frequency hopping communication system synchronization[EB/OL]. http://www.Scientific.net/AMR.403-408.59.

[9] PHILIPP S, ROGER W. Gradient clock synchronization in wireless sensor networks[A]. Proceedings of the 2009 International Conference on Information Processing in Sensor Networks[C]. San Francisco, USA. Piscataway, NJ, USA, 2009. 37-48.

[10] ARUN K T, AJAY A, YASHPAL S. An approach towards secure and multihop time synchronization in wireless sensor network[J]. Communications in Computer and Information Science, 2010, 101(2): 297-302.

[11] GAL B, HERZBERG A, KEIDAR L. Keeping denial-of-service attackers in the dark[J]. IEEE Transactions on Dependable and Secure Computing, 2007, 4(3): 191-204.

[12] LIN K, JIA C F, WENG C. Distributed timestamp synchronization for end hopping[J]. China Communications, 2011, 8(4): 164-169.

作者简介：

林楷（1985-），男，江西上饶人，南开大学博士生，主要研究方向为网络和信息安全。

贾春福（1967-），男，河北文安人，博士，南开大学教授、博士生导师，主要研究方向为信息安全与可信计算、恶意代码发现与分析。

石乐义（1975-），男，山东临朐人，博士，南开大学副教授、硕士生导师，主要研究方向为计算机网络、网络安全、动态蜜罐和博弈理论。