一种基于安全网络编码技术的抗污染攻击机制

王海鹏,梅中辉
(南京邮电大学 通信与信息工程学院,江苏 南京 210003)

摘 要:文中主要研究在安全网络编码技术中的抗污染问题。由于网络编码会在中间节点生成新的数据包,所以只要注入少量污染数据包就会造成污染扩张,从而造成网络性能下降。所以有必要在中间节点验证网络中是否发生污染攻击,但同时又要减少检验所带来的时间延迟。文中提出一种自适应抗污染攻击机制。该机制可以针对污染攻击严重情况来自适应调节中间节点的检测步骤,同时提高了运算效率并降低了验证时延。仿真结果表明,理论结果和实际计算吻合,自适应验证机制的时延相比传统验证机制较小。

关键词:安全网络编码;污染攻击;自适应验证;同态
中图分类号:TP301 **文献标识码**:A **文章编号**:1673-629X(2016)07-0094-06
doi:10.3969/j.issn.1673-629X.2016.07.020

A Scheme against Pollution Attacks Based on Secure Network Coding

WANG Hai-peng, MEI Zhong-hui
(College of Telecommunication & Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: It mainly deals with pollution attacks in secure network coding in this paper. The new packet will be produced in the intermediate nodes by employing network coding, so a small amount of pollution data packet could cause pollution expansion, which results in substantially degraded performance for the network. Therefore, it is necessary to verify whether the intermediate nodes are in the pollution attack or not, and at the same time to reduce the time delay caused by security detection. An adaptive mechanism that can dynamically adjust the authentication strategy of intermediate nodes is proposed so as to improve operational efficiency and decrease the time delay brought by verification. Simulation shows that theoretical results are in accordance with actual calculation, and the time delay of the adaptive mechanism is less than that of traditional verification mechanism.

Key words: secure network coding; pollution attack; adaptive verification; homomorphic

0 引 言

网络编码[1]技术不同于传统存储-转发技术,中间节点可以对数据包进行计算生成新的数据包。网络编码可以显著提高网络吞吐量、减小网络能耗[2-3]等,但是安全问题是网络编码不能回避的问题,主要存在两种攻击方式:窃听攻击[4-6]和污染攻击[7-8]。其中污染攻击所造成的影响更严重,一个污染包注入到网络中会在其他节点与别的未污染数据包组合生成新的数据包,导致新生成的数据包也受到污染,会造成污染的扩散,浪费了网络资源。

目前对于污染攻击存在两种解决方案:

(1)基于信息论[9-10]的解决方案。通过给数据包增加冗余比特以达到检验目的,但只能在目的节点才能检验出污染包

(2)基于密码学[11-13]的解决方案。这种方式可以使网络中每个节点都检验出污染包,可以有效抑制污染数据包的传播。

文中采用基于密码学的方案,但是传统的密码学解决方案存在计算量过大的特点,因此并不实用。He Ming 等[14]提出一种应对污染攻击的自适应验证机制 Adapkeys,但网络需要严格时间同步。E. Kehdl 等[15]提出的 Null key 制度利用正交性来验证数据包是否受到污染,但对每一代数据包的正交向量的分发与计算都比较复杂。Zhang Peng 等[16]提出一种基于正交子

收稿日期:2015-10-19 **修回日期**:2016-01-22 **网络出版时间**:2016-06-22
基金项目:国家科技重大专项(2010zx03003-003)
作者简介:王海鹏(1991-),男,硕士研究生,研究方向为安全网络编码;梅中辉,副教授,研究生导师,研究方向为网络编码技术、协助通信技术等。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20160622.0842.008.html

空间的验证机制,计算量较小,但不能区分随机错误和污染攻击。刘济恺[17]在 He Ming 等研究结果的基础上提出一种自适应的验证机制,该方案能对随机错误和污染攻击进行区分,但签名验证机制牵扯到大量的模指数运算,计算量较大。

结合上述机制,文中提出一种基于正交子空间原理的自适应验证机制,可相应减小算法的计算复杂度。

1　系统模型与攻击者模型

考虑一个多播网络,信源 S 想发送 $x_1, x_2, \cdots, x_m$ 给多个接收者集合 $\{R_i\}$,每个数据包 x_i 都是一个 $m+n$ 维的向量,即 $x_i = (\overbrace{0\cdots0}^{i-1}, 1, \overbrace{0, \cdots, 0}^{m}, x_{i,1}, x_{i,2}, \cdots, x_{i,n})$,且每个符号都属于有限域 F_p,网络采用随机网络编码方式。

假设只有信源节点 S 和接收者是可信的,而中间节点都是不可信的,可能发动污染攻击。而攻击者试图注入一些污染数据包 w 进入网络时,即 $w \notin \text{span}\{x_1, x_2, \cdots, x_m\}$,攻击者(adversary nodes)通过收集网络合法数据包,并试图使数据包 w 通过其他节点(innocent nodes)的验证。此外也假设中间节点可能会共谋(collude),即中间节点中的攻击者可能会相互协作发动攻击,在该网络中假设攻击者的计算能力有限。

2　验证机制及原理

尽管数据包在网络中会经过很多轮的随机网络编码,但是信源数据包所组成的(linear subspace)线性子空间 V 是不变的,可以通过检测接收到的数据包 w 是否属于这个子空间 V($w \in V$)来判断是否为原始数据包。网络的中间节点可以从 V 的正交空间选出一个向量 $\boldsymbol{v}^{\mathrm{T}}$ 来描述向量空间 V,如果 $\boldsymbol{w} \cdot \boldsymbol{v}^{\mathrm{T}} \neq 0$,则 $\boldsymbol{w}$ 不是原始数据包。但是每代数据包都需要分别计算 V 的正交分量并将其分发至网络中的中间节点,导致算法复杂且开销大。文献[16]提出相应解决方案,通过给每个数据包加上若干标签与签名,使加过标签或签名的数据包 $\boldsymbol{w}'$ 与 $\boldsymbol{v}^{\mathrm{T}}$ 内积为 0,这样就不用频繁发送 $\boldsymbol{v}^{\mathrm{T}}$,节省了发送带宽。基于这一基本原理,文中提出了 Macsig 机制,具有如下特点:

(1)有效应对污染攻击,特别是对于对称公钥机制中特有的一种污染-标签攻击(tag pollution)能做到有效应对。

(2)相对于其他基于 MAC[12] 的验证机制能节省带宽,采用提出的 Double-Random Key Distribution 机制,需要的标签(tag)更少。

(3)计算更加高效,验证时延小。

但是 Macsig 机制没有对随机传输错误与污染攻击进行区分对待,前者更容易检测出来,而不需要复杂的计算。而且通常网络规模很大,小规模的数据包污染是可以忍受的,在这里结合文献[14]的自适应框架,提出一种新的验证机制。

2.1　参数设置

T:节点工作参数,每个节点均独立维护,系统初始化时,每个节点的 $T=0$;

d:每个数据包所带有的一个参数,代表经过上次验证后所经过的跳数,最小值为 0,最大值为 $d_{\max}$;

$T_{\max}$:可以调节的安全参数,代表 T 的最大值;

Flag:每个节点单独维护的安全参数,代表当前节点是否有污染数据包。为 0 代表安全,为 1 代表检测状态,处于检测状态的节点不参加网络编码,默认初始化为 0;

ΔT:安全参数,具体 $\Delta T = \gamma(1 - \frac{d_{\min}}{d_{\max}})$。其中,$\gamma$ 是可以根据网络情况来调节的,$d_{\min}$ 代表所有没有通过 Macsig 机制的数据包中 d 的最小值;

id:代表当前数据包属于哪一代,$\text{id} \in F_p$;

G:一个阶数为 p 的乘法群 G,乘法群的生成元 g,p 是一个位数很大的质数;

PRG:伪随机数生成器(pseudorandom generator),输入一个种子生成 $m+n+l+2$ 维随机向量,$F_p \Rightarrow F_p^{m+n+l+2}$,每个节点都拥有相同的 PRG;

β:随机生成一个私钥 $\beta = (\beta_1\beta_2\cdots\beta_{m+l+1}) \Leftarrow F_p^{m+l}F_p$,一个 $m+l+1$ 维向量;

h:公钥,计算公式 $h = g^{\beta} = (g^{\beta_1}\cdots g^{\beta_{m+l+1}})$;

K:一个 MAC 密钥池,每个密钥 γ_i 都是一个 $m+n+1$ 维的向量,应满足下列格式:$\gamma_i = \{\gamma_{i,1}\cdots\gamma_{i,m+n+1}\} \Leftarrow F_p^{m+n}F_p$。

2.2　签名与验证步骤

1)源节点签名。

对于信源 S 发出的 m 个 $x_1, x_2, \cdots, x_m$ 消息,生成唯一的代标识符 id。MAC Key 选取方法同文献[16]一样,信源节点从密钥集合 K 中随机选出一个大小为 l 的密钥子集,作为验证该代数据包的 MAC Key 集合,然后在该代的每个数据包中附上这 l 个密钥的对应索引。网络中其他节点分配密钥的方式同文献[18]。

源节点签名过程可归纳如下:

(1)生成随机校验值:把每一代 id 作为种子放入 PRG 后,生成一个随机向量 $\text{PRG}(\text{id}) = R = \{r_1, r_2, \cdots, r_{m+n+l+2}\}$,对于 x_i 计算校验值 z_i:

$$z_i = \sum_{j=1}^{m+n+l+2} r_j x_{i,j} (\text{mod} p) \tag{1}$$

(2) 对于数据包 x_i，加上 l 个 MAC 标签 $\{t_{i,1}\cdots t_{i,l}\}$，其中 $t_{i,j}=\dfrac{-\sum_{r=1}^{m+n}\gamma_{j,r}x_{i,r}}{\gamma_{j,m+n+1}}$。

(3) 为数据包 x_i 加上签名 σ_i，并且 $\sigma_i=\dfrac{-(\sum_{j=1}^{m}\beta_j x_{i,j}+\sum_{j=1}^{l}\beta_{j+m}t_{i,j})}{\beta_{m+l+1}}$，最后将 l 个标签 $t_{i,j}$、代标识符 id、参数 d、签名 σ_i 与随机校验值放在数据包中，将 d 设置为 0，其中 id 与 d 不参与编码操作。

2) 中间节点验证。

当中间节点收到一个编码向量 $\boldsymbol{w}=(w_1,w_2,\cdots,w_{m+n})$，其附带的其他消息为 $(t_{w,1},t_{w,2},\cdots,t_{w,l},\sigma_w,z_w,\mathrm{id},d)$，验证步骤如图 1 所示。

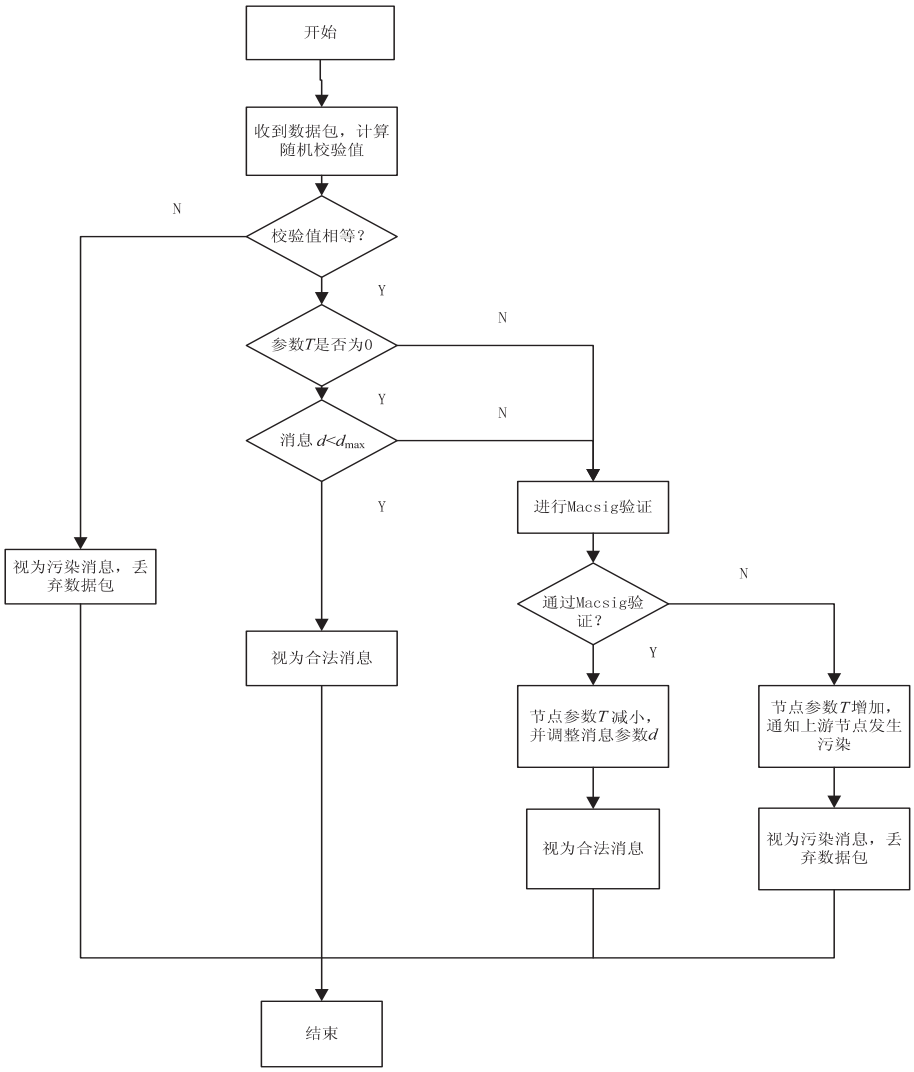

图 1 中间节点验证流程

(1) 校验值验证。主要针对随机发生的错误，根据 id 获得随机校验向量 PRG(id) = $R=\{r_1,r_2,\cdots,r_{m+n+l+2}\}$，然后计算校验值。如果 $z_w=z$，则通过校验值验证，视为合法消息，否则为污染消息。而对于同一代数据包，计算 R 后并存储，下次直接调用即可。

(2) Macsig 验证机制，通过校验值验证后，中间节点判断参数节点 T 和消息附带的经过跳数 d_w，只有满足 $d_w<d_{\max}$ 且 $T=0$，则消息不需要进行 Macsig 验证，认为是合法消息，否则进行 Macsig 验证。

Macsig 验证步骤可归纳如下：

步骤 1：为避免 tag-pollution，计算 $\delta=\prod_{i=1}^{m}h_i^{w_i}\prod_{i=1}^{l+1}h_{m+i}^{w_{m+n+i}}$，若等于 1 继续，否则丢弃。

步骤 2：计算对于当前中间节点所拥有的 MAC Key 集合 K'（$K'\subset K$）中的每一个 γ_i 密钥，判断 $\varepsilon_i=\sum_{r=1}^{m+n}\gamma_{i,r}w_r+\gamma_{i,m+n+1}t_{w,i}$ 取值是否等于 0，若所有对应标签 $\varepsilon_i=0$，则该消息 w 是合法消息，否则是污染消息，应当丢弃。

步骤 3：若是合法消息，则将该消息的经过跳数 d_w 设置为 0，若当前中间节点的 $T>0$，则 T 减小 1，否则 T 仍为 0。

当数据包 w 为污染包，当前节点的 T 增加 ΔT，当

$T + \Delta T > T_{\max}$,则将 T 设置为 $T_{\max}$,发送警告给上游节点,且当前节点的 FLAG = 1,对当前节点进行批验证。然后上游节点在收到警告后,先验证警告的真伪性,如果通过验证,上游节点将对自己缓存中的数据包进行批验证(即利用缓存数据包随机生成一个新包,并检测新包是否受到污染)。若通过批验证,此上游节点设 FLAG 保持不变仍为0。若不通过,首先设置此上游节点的 $T = T + \Delta T$,并且将此上游节点的 FLAG 变为1,此上游节点将停止产生新的数据包,进入隔离状态。在隔离状态期间进行二分法查找污染数据包,直到从缓存中清除污染数据包,FLAG 重新归0,隔离状态结束,此上游节点将重新参加网络编码。

步骤4:中间节点对验证通过的数据包进行编码输出,将消息的校验值、签名、标签值与消息一起进行相同的编码操作。假设编码 c 个合法消息 $w_1, w_2, \cdots, w_c$ 所对应的校验值、签名和标签为 $(z_{wi}, \sigma_{wi}, t_{wi,1} \cdots t_{wi,r})$,编码系数为 $a_1 \cdots a_c$,则编码过程为:

$$w' = \sum_{i=1}^{c} a_i w_i (\mathrm{mod}p), z_w' = \sum_{i=1}^{c} a_i z_{wi} (\mathrm{mod}p), \sigma_w' = \sum_{i=1}^{c} a_i \sigma_{wi} (\mathrm{mod}p), t_{w',i} = \sum_{j=1}^{c} a_j t_{wj,i} (\mathrm{mod}p) \quad (2)$$

新数据包 d 值更新为:

$$d_w' = \max(d_{w_1}, d_{w_2}, \cdots, d_{w_c}) + 1 \quad (3)$$

3)接收节点接收验证。

目的节点收到数据包 w 后,先进行校验值验证,若不通过,直接结束。通过后进行 Macsig 验证,通过则认为是合法数据包,放入接收缓存,等待解码操作,否则丢弃数据包。

2.3　验证方式正确性

现在需证明无论是校验值、MAC 校验值还是签名都满足同态性。

假定信源节点 S 的第 id 代的原始数据包 $x_1, x_2, \cdots, x_m$,所对应的合法签名为 $(\sigma_1, \sigma_2, \cdots, \sigma_m)$,生成的随机校验值为 $(z_1, z_2, \cdots, z_m)$,生成的标签为 $(t_{1,1}, \cdots, t_{1,l}, t_{2,1}, \cdots, t_{2,l}, \cdots, t_{m,1}, \cdots, t_{m,l})$。假设中间节点收到的数据包为 w',对应签名为 σ_w',校验值为 z_w',对应的标签为 $(t_1', t_2', \cdots, t_m')$,代标识符为 id,编码系数为 $a_1, a_2, \cdots, a_m$。

校验值满足同态性:

$$z_w' = \sum_{j=1}^{m+n+l+2} r_j w_j' \mathrm{mod}p = \sum_{j=1}^{m+n+l+2} r_j \sum_{i=1}^{i=m} a_i x_{i,j} \mathrm{mod}p = \sum_{i=1}^{m} a_i z_i \mathrm{mod}p \quad (4)$$

签名的同态性:

$$\delta = \prod_{i=1}^{m} h_i^{w_i'} \prod_{i=1}^{l+1} h_{m+i}^{w'_{m+n+i}} = g^{\sum_{i=1}^{m} (\beta_i w_i') + \sum_{i=1}^{l+1} (\beta_{m+i} w'_{m+n+i})} = g^{\sum_{i=1}^{m} (\beta_i \sum_{d=1}^{m} a_d x_{d,i}) + \sum_{i=1}^{l+1} (\beta_{m+i} \sum_{d=1}^{m} a_d x_{d,m+n+i})} = 1 \quad (5)$$

MAC 标签的同态性:

$$\varepsilon_i = \sum_{r=1}^{m+n} \gamma_{i,r} \sum_{d=1}^{m} a_d x_{d,r} + \gamma_{i,m+n+1} \sum_{d=1}^{m} a_d t_{d,i} = \sum_{d=1}^{m} a_d \Big(\sum_{r=1}^{m+n} \gamma_{i,r} x_{d,r} + \gamma_{i,m+n+1} t_{d,i} \Big) = 0 \quad (6)$$

对于数据包中的参数 d 不需要保护。如果 d 被篡改变大,那么受污染数据包会更快达到 $d_{\max}$ 而被检测。如果参数 d 被改小,那么当被发现后,根据 $\Delta T = \gamma(1 - \frac{d_{\min}}{d_{\max}})$,该节点会保持更长时间的 $T > 0$ 状态。所以该节点会在较长时间内经历更加严格的检测,污染会被检测出来。

Macsig 机制的安全性基于中间节点不知道 MAC key $\{\gamma_i\}_{i=1}^{l}$ 和 secret key $\{\beta_1 \cdots \beta_{m+l+1}\}$,但如果中间节点收集足够多的合法数据包,那么它就可能解出上述参数,导致机制失效。所以当信源节点发送 $\mathrm{Min}(m + n + 1, m + l + 1)$ 个数据包以后,为了安全起见,信源更新 MAC key $\{\gamma_i\}_{i=1}^{l}$ 和 secret key $\{\beta_1 \cdots \beta_{m+l+1}\}$,并重新分配密钥。

该机制还有一个漏洞,即如果有多于预设值 c 的中间节点预谋,则攻击节点可能会生成污染数据包,并通过其他节点的验证。为了解决这个问题,信源应当对整个文件计算 Hash 值,并发送给信宿,这样信宿节点在解码成功后交给上层前,应当再计算 Hash 值。将两个值进行比较,如果相同,认为未发生污染;如果不同,则认为该机制失效,考虑用其他更高安全性的抗污染机制。

2.4　验证警告机制

中间节点在检测到污染数据包后,会向上游节点发送警告消息,如果不对警告消息进行验证,就可能会有恶意节点不断发送伪造警告,导致上游节点不断浪费资源去验证是否存在污染。采用类似文献[19]中的机制,来降低验证警告消息的时间开销。在信源发送数据前,信源给每个中间节点分发一个唯一的随机掩码(random-masks)和基于掩码的随机唯一的校验和(mask-based Checksums)。随机掩码定义为 $\vec{t} = (t_1, t_2, \cdots, t_m)$,$t_m \in F_p$,$m$ 代表数据包的长度。然后对每个信源消息 x_i,分别计算校验和 $f(x_i) = \sum_{k=1}^{m} t_k x_{i,k}$,然后获得一个向量 $\vec{f} = \{f(x_1), f(x_2), \cdots, f(x_n)\}$。

在信源发送数据前,每个中间节点可以通过保密信道先下载 $(\vec{t}, \vec{f})$,每个中间节点得到的都不一样,并互相保密。当收到警告信息时,此节点先利用缓存

数据包,随机线性编码生成一个新的数据包 e, $e=\sum_{i=1}^{n} c_i x_i$。验证下列公式是否成立:

$$f(e)=\sum_{k=1}^{m} t_k e_k=\sum_{k=1}^{m} t_k\left(\sum_{i=1}^{n} c_i x_{i,k}\right)=\sum_{i=1}^{n} c_i\left(\sum_{k=1}^{m} t_k x_{i,k}\right)=\sum_{i=1}^{n} c_i f(x_i) \quad (7)$$

如果式(7)成立,则判断该警告是伪造的,即没有发生污染。若不成立,则认为发生了污染,即警告消息是有效的(Valid)。

3 仿 真

由于用到了双重随机密钥[16]分配,每个数据包都需要包括 l 个 MAC 标签、l 个 MAC 所对应的密钥索引、代标识符、签名和随机校验值。和信息符号一样,签名、校验值、MAC 标签、代标识符都定义在 F_p 上。根据文献[16],有限域尺寸都定义为 $|p|=\lceil \log_2 p \rceil=128$ bit,而 MAC key 索引尺寸采用 $\lambda=32$ bit。根据文献[16]定理5,每个数据包一共有 l 个 MAC 标签用于验证 $(l=\frac{1}{1-\delta}\mathrm{e}(c+1)\ln\frac{1}{\varepsilon})$。其中,$\delta=0.1$, $n=20m$, $\Pr=1-\varepsilon$,参数 c 代表参与共谋的节点个数,参数 Pr 代表双随机密钥分配机制(double-random key distribution)能成功抵御 c 个攻击节点共谋攻击的概率。参数 Pr 与 c 越高,相应安全等级越高。所以新的机制的带宽负载计算公式应为:

$$Q=\frac{(l+3)|p|+\lambda l}{(m+n)|p|+(l+3)|p|+\lambda l}$$

仿真结果如图 2 所示。

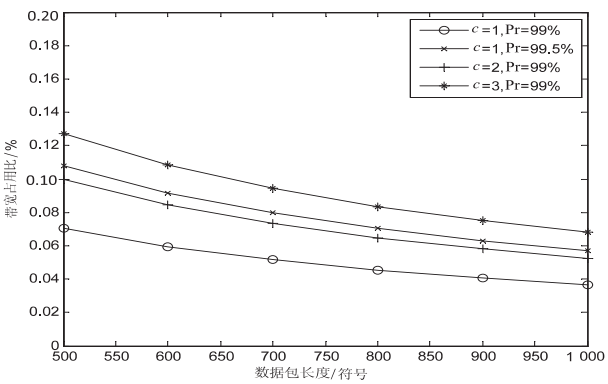

图 2 系统带宽开销

从图中可以看出,随着数据包长度的增加,带宽负载是呈下降趋势的。参与共谋攻击节点越多,带宽占用越大。

由于有限域上的模指数运算非常花时间,所以主要考虑节点在验证时的计算耗时。利用 NTL 函数库,可以得出在验证阶段的随机校验阶段需要 $(m+n+l+2)$ 次乘法,$(m+n+l+1)$ 次加法,而在 Macsig 阶段则进行了 $(m+l+1)$ 模指数运算,$(m+l)+l*(m+n+1)$ 次乘法和 $(m+n)$ 次加法,其他参数设置和上面一样。而文中提出机制的优点在于不是每一个节点都完全执行所有的验证步骤。假设 $d_{\max}=5$,即污染数据包最多在网络中传播 5 跳,那么一个合法数据包在最理想的情况下要经历 5 次随机校验值检测和一次的 Macsig 认证,而文献[11]提出的基于同态签名的传统方案,则要经历 5 次一样的校验。每次验证都至少进行 $(m+n)$ 次模指数运算和 RSA 解密步骤。文献[17]的方案同样运用了自适应验证的思想,但每次验证都同样需要至少 $(m+n)$ 次模指数运算。

方案仿真结果如图 3 所示。

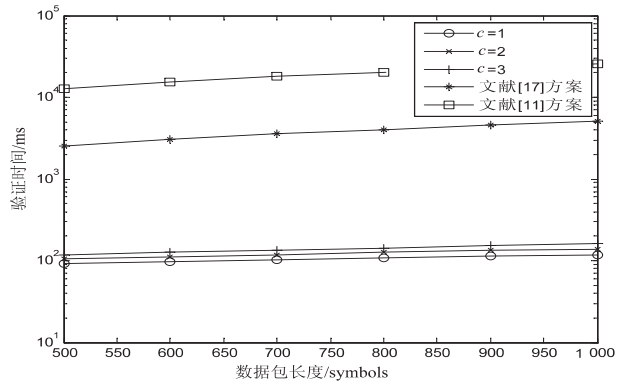

图 3 系统计算复杂度开销

从图中可以看出,采用自适应验证思想的方案都比文献[11]方案验证时延小,而且文中机制相比同样采用自适应思想的文献[17]的延迟要小,是因为文中方案降低了模指数运算次数。如果考虑极端情况,即数据包每一跳都恰好满足检测条件,需要被检测,根据仿真结果的趋势,可以看出计算量仍然较低,不会超过其他机制。当安全参数 c 越小时验证越快,但付出的代价是协议安全性降低,抗共谋攻击的能力降低,而且新的机制需要复杂的密钥分配机制来保证机制的可行性。

4 结束语

文中主要介绍了一种能自适应网络情况的抗污染攻击方案。传统的污染检测方案计算量大,验证耗时,且不能灵活改变安全检测等级。一方面将自适应框架引入到检测方案中,另一方面将方案中的检测机制予以改进,减小了模指数运算的次数。仿真结果表明,与传统的检测机制相比,有效降低了检测时延。

参考文献:

[1] Ahlswede R, Cai N, Li S Y R, et al. Network information flow [J]. IEEE Transactions on Information Theory, 2000, 46(4): 1204-1216.

[2] 杨 林,郑 刚,胡晓惠. 网络编码的研究进展[J]. 计算机

研究与发展,2008,45(3):400-407.
[3] 黄　政,王　新. 网络编码中的优化问题的研究[J]. 软件学报,2009,20(5):1349-1361.
[4] 俞立峰,杨　琼,于　娟,等. 防窃听攻击的安全网络编码[J]. 计算机应用研究,2012,29(3):813-818.
[5] Cai Ning,Chan T. Theory of secure network coding[J]. IEEE Journals & Magazines,2011,57:416-423.
[6] 周业军,李　晖,马建峰. 一种防窃听的随机网络编码[J]. 西安电子科技大学学报:自然科学版,2009,36(4):696-701.
[7] 曹张华,唐元生. 安全网络编码综述[J]. 计算机应用,2010,30(2):499-505.
[8] 张盛勇,陈世康. 网络编码的安全问题初探[J]. 通信技术,2012,45(1):105-107.
[9] Ho T,Leong B,Koetter R,et al. Byzantine modification detection in multicast networks using randomized network coding[C]//Proc of IEEE international symposium of information theory. [s. l.]:IEEE,2004.
[10] Jaggi S,Langberg M,Katti S,et al. Resilient network coding in the presence of byzantine adversaries[C]//Proc of international conference on computer communications. [s. l.]:[s. n.],2007.
[11] Yu Z,Wei Y,Ramkumar B,et al. An efficient signature-based scheme for securing network coding against pollution attacks[C]//Proc of international conference on computer communications. [s. l.]:[s. n.],2008.
[12] Agrawal S,Boneh D. Homomorphic MACs:MAC-based integrity for network coding[C]//Proc of international conference on applied cryptography and network security. [s. l.]:[s. n.],2009.
[13] Zhao F,Kalker T,Medard M,et al. Signatures for content distribution with network coding[C]//Proc of IEEE international symposium on information theory. [s. l.]:IEEE,2007.
[14] He Ming,Chen Lin,Wang Hong,et al. Adapkeys:an adaptive security scheme for network coding[C]//Proc of IEEE Asia-Pacific services computing conference. Guilin:IEEE,2012:309-314.
[15] Kehdl E,Li B. Null keys:limiting malicious attacks via null space properties of network coding[C]//Proc of international conference on computer communications. [s. l.]:[s. n.],2009.
[16] Zhang Peng,Jiang Yixin,Lin Chuang,et al. Padding for orthogonality:efficient subspace authentication for network coding[C]//Proc of international conference on computer communications. Shanghai:IEEE,2011:1026-1034.
[17] 刘济恺. 防污染的安全网络编码研究[D]. 成都:西南交通大学,2014.
[18] Canetti R,Garay J,Itkis G,et al. Multicast security:a taxonomy and some efficient constructions[C]//Proc of international conference on computer communications. [s. l.]:[s. n.],1999.
[19] Gkantsidis C,Rodriguez P. Cooperative security for network coding file distribution[C]//Proc of international conference on computer communications. [s. l.]:[s. n.],2006.

(上接第 93 页)

分重要,动态截断可疑网络数据流。根据安全态势做智能化网络防御。基于进程和网络元组的网络流量和连接控制,基于并行算法的协同安全组件,研究了提高云计算网络的安全机制的各种方法。

参考文献:

[1] 姜　伟,方滨兴,田志宏,等. 基于攻防博弈模型的网络安全测评和最优主动防御[J]. 计算机学报,2009,32(4):817-827.
[2] 胡　平,李　臻,彭纪奎. 基于入侵检测的分布式防火墙的应用研究[J]. 微电子学与计算机,2011,28(6):126-130.
[3] Russinovich M,Solomon D A. Windows internals:including Windows server 2008 and Windows Vista[M]. [s. l.]:Microsoft Press,2009.
[4] 马兆丰,顾　明,孙家广. 基于角色的可信数字版权安全许可授权模型[J]. 清华大学学报:自然科学版,2006,46(4):534-538.
[5] 吴　涛,张毛迪,陈传波. 一种改进的统计与后串最大匹配的中文分词算法研究[J]. 计算机工程与科学,2008,30(8):79-82.
[6] 李鸿彬,林　浒,侯辉超,等. 一种 SIP 分布式洪泛攻击的减弱方法[J]. 小型微型计算机系统,2012,33(5):995-999.
[7] 胡建理,周　斌,吴泉源,等. P2P 网络环境下基于信誉的分布式抗攻击信任管理模型[J]. 计算机研究与发展,2011,48(12):2235-2241.
[8] 陈伟东,王　超,张　力,等. 服务器系统安全内核研究与实现[J]. 计算机应用与软件,2013,30(3):304-307.
[9] 陈伟东. 一种 HTTP 通信内容检测的方法:中国,CN104022924A[P]. 2014-09-03.
[10] Ficara D,Giordano S,Procissi G,et al. An improved DFA for fast regular expression matching[J]. ACM SIGCOMM Computer Communication Review,2008,38(5):29-40.
[11] 亓亚烜,李　军. 高性能网包分类理论与算法综述[J]. 计算机学报,2013,36(2):408-421.
[12] 陈伟东,张　力. Windows Rootkit 分析与检测综合方法[J]. 信息化纵横,2009(12):10-15.
[13] Modi C,Patel D,Borisaniya B,et al. A survey of intrusion detection techniques in cloud[J]. Journal of Network and Computer Applications,2013,36(1):42-57.