doi:10.3969/j.issn.1000-436x.2013.11.010

抗污染攻击的自适应网络编码传输机制

何明，邓罡，王宏，龚正虎
(国防科学技术大学 计算机学院，湖南 长沙 410073)

摘 要：研究了网络编码中的污染攻击问题，提出了一种抗污染攻击的自适应网络编码传输机制 ASNC (adaptive secure network coding)。在编码数据分组的传输过程中，该机制利用网络编码的时间和空间特性有效控制污染数据分组的传播。同时，ASNC 机制创新性地促使网络编码系统动态调整安全策略，自适应于当前网络安全态势。此外，为了达到更好的实用性，ASNC 机制有效利用网络编码的编码空间特性，不需要额外的安全数据通道和数据分组加密操作。ASNC 机制的安全分析和仿真结果表明，其能够有效抵抗污染攻击，与不具有自适应能力的机制相比具有更好的安全效率。

关键词：网络编码；安全；污染攻击；时空特性；自适应
中图分类号：TP 393 **文献标识码：**A **文章编号：**1000-436X(2013)11-0081-11

Adaptive secure network coding scheme against pollution attacks

HE Ming, DENG Gang, WANG Hong, GONG Zheng-hu
(College of Computer, National University of Defense T hnology, Changsha 410073, China)

Abstract: The problem of pullution attacks was focused on and ASNC (adaptive secure network coding) scheme was proposed, which is an adaptive security scheme against pollution attacks in network coding systems. The proposed scheme allows participating nodes to detect polluted packets based on time and space properties of network coding. It is an innovative security scheme which can dynamically adjust the authentication strategy of participating nodes according to the security situation. In addition, ASNC scheme provides an efficient packet authentication without requiring the existence of any extra secure channels. Security analysis and simulation of the scheme were also and the results demonstrate the practicality and efficiency of the ASNC scheme.

Key words: network coding; security; pollution attack; time and space properties; adaptive

1 引言

图论中最大流最小割定理[1]明确指出：通信网络端到端的最大信息流是由网络有向图模型的最小割决定。由于传统网络的组播传输首先需要构建一棵组播树来确定组播传输路径，但是组播树的建立问题是 NP 完全问题，所以网络最大信息流的理论上界，在很长一段时间里显得遥不可及。网络编码理论由 AHLSWEDE 等[2]首次提出，它创造性地将编码和路由有机地融合为一体，建立了一种全新的网络体系结构，解决了组播路由构造这一经典难题。同时，AHLSWEDE 等人详细阐述了网络编码相对于传统存储转发模式的优势，指出网络编码可以达到组播网络最大流传输的理论上限。由于对网络通信能力的提升，网路编码已经作为关键技术用于无线传感器网络[3]、移动通信网络[4]以及 P2P 内容分发网络[5]等。此外，线性网络编码[6]技术能使中间节点处理的消息都是原始消息向量的编码组合，这实际上是对原始消息的一种信息隐藏。在恶意节点只能窃取原始消息部分线性无关编码向量的前提下，攻击者就无法解码恢复出原始消息。文献[7,8]有效利用了网络编码的这种保密特性，提出防止恶意节点窃听行为的具体方法。

伴随着网络编码技术的发展，其在实际应用中

收稿日期：2013-01-05；**修回日期：**2013-04-07
基金项目：国家高技术研究发展计划(“863”计划)基金资助项目(2011AA01A103)
Foundation Item: The National High Technology Research and Development Program of China (863 Program) (2011AA01A103)

存在的安全问题也为很多研究人员所关注，安全网络编码逐渐成为国内外学者的一个研究热点。在网络编码系统中，攻击者向传输网络注入错误或伪造的信息，扰乱网络的正常通信，使目的节点无法得到原始信息。这种攻击方式被称为污染攻击。由于网络编码的信息融合特性，污染攻击对网络编码的危害尤其严重。因为所有中间节点都通过对接收信息进行编码组合操作产生分发信息，所以只要一个节点的某个接收信息被污染了，其所有分发的信息都是污染信息。这就使污染信息能迅速扩散，也许仅仅是一个污染数据分组就能影响一片网络，甚至使整个网络编码系统陷入瘫痪状态。污染攻击消耗了网络带宽和节点处理能力等网络资源，并且极大地影响了网络的可用性，因此设计合理的污染检测机制，有效过滤污染信息对网络编码系统来说尤为重要。

目前，已经有很多研究学者针对网络编码的污染攻击问题提出相应安全机制，这些机制可以分成以下 2 类。

1) 端到端检测机制[9~11]：发送端和接收端负责检测污染信息，中间节点不需要验证信息的合法性直接编码传输所收到的信息。端到端检测机制仅对现有网络编码模型做最小的改动，不增加网络中间节点的计算负担。同时，中间节点仅负责简单的计算操作，符合网络编码分布式低复杂性的初衷。但当攻击者选择网络的瓶颈链路作为污染攻击的目标时，端到端检测机制将束手无策。此外，端到端纠错机制一般对网络中恶意节点的数量或者比例、被窃听的链路数量以及被篡改的消息数量等都做了限制性的假设，所以端到端纠错机制的抵抗污染攻击能力十分有限。

2) 网络检测机制：网络中所有诚实节点利用相关安全加密信息验证所接收信息的合法性。网络检测机制相对于端到端检测机制，可以尽可能早地过滤出污染信息，从而达到更好的传输效率和安全性。但是，现有的网络检测机制要么不具备足够的安全能力抵抗多攻击者的高度污染，要么将导致网络性能的大幅度下降。例如 ELIAS 等[12]提出的 Null keys 机制，其根据合法数据向量构成一个子空间，源节点将该子空间的正交向量提前分配给中间节点，中间节点通过判断接收到的数据向量是否与正交向量正交来判断是否存在污染数据分组。可是当攻击者能力比较强时，攻击者的邻居合法节点所分配的正交向量很容易被攻击者截取，这时攻击者可以轻松构造出能通过 Null keys 机制验证的污染数据分组。而在面临多个这样的攻击者时，Null keys 机制根本无法应付。而同态散列方法由 KROHN 等[13]首次提出，其能让中间节点有效检测出污染数据分组。但是同态散列操作计算复杂度比较高，而且中间节点必须使用同态散列操作验证每一个接收到的编码数据分组，所以同态散列方法非常耗费计算资源，网络传输的性能也就受同态散列操作的计算效率所限制。除了安全能力和运行效率之间的问题外，现有的网络检测机制在没有攻击发生的情况下也要消耗大量的网络计算资源和带宽资源不断检测编码数据分组，缺乏实际应用所必需的弹性。

本文设计了一种网络编码传输机制 ASNC，其能高效抵抗污染攻击。ASNC 能动态调整中间节点检测策略，并自适应于当前网络安全态势。当污染攻击严重时，ASNC 自动实现严格的检测策略，当网络不存在污染攻击时，ASNC 可以实现宽松的检测策略以达到高效的数据传输。

2 系统模型攻击模型

本节简要介绍网络编码的系统模型和攻击模型，这是 ASNC 机制的基础。

2.1 系统模型

本文提出的网络模型是一个非循环有向图 $\langle V,E\rangle$，网络中允许 2 个节点之间存在多条信道，V 是节点集合，E 是边集合，每条边表示单位时间单位消息的通信信道。边 $e=(v,v')\in E$ 的头部和尾部分别用 $v=head(e)$ 和 $v'=tail(e)$ 表示。在节点 $v\in V$ 结束的边集为 $GI(v)=\{e\in E: head(e)=v\}$，出发的边集表示为 $GO(v)=\{e\in E: tail(e)=v\}$。

认为网络包含一个源节点 $S\in V$，多个目的节点 $R\subset V, R=\{R_1,R_2,\cdots,R_k\}$ 以及一系列中间节点。目的节点也能作为中间节点编码转发数据分组。S 把数据分组切分为代（generation），关注当前编码和传输的一代，只有同一代的信息才进行相互编码。每一代由 n 个数据块组成，每一个数据块可以表示成有限域 F_q 里的 m 个元素，q 是一个素数。每个数据块 $\vec{p}_i$ 是由 m 个有限域 F_q 的元素组成的向量为

$$\vec{p}_i=(p_{i1},p_{i2},\cdots,p_{im}),\quad p_{ij}\in F_q, 1<j<m \tag{1}$$

一代数据 G 由 n 个数据块组成，其可表示成一

个矩阵

$$G=[\vec{p}_1,\vec{p}_2,\cdots,\vec{p}_n]^{\mathrm{T}} \tag{2}$$

源 S 生成消息编码向量 $\vec{e}=\vec{c}\boldsymbol{G}=\sum_{i=1}^{n}c_i\vec{p}_i$，其中，$\vec{x}_i$、$c_i(1\le i\le n)$ 是有限域 F_q 里的随机元素。然后 S 发送数据分组 $(\vec{c},\vec{e})$。将 $(\vec{c},\vec{e})$ 称为编码分组，$\vec{c}$ 为全局编码向量。中间节点收到编码分组，通过计算这些编码分组的线性组合，形成新的编码分组，并发送到后续网络。当某个目的节点接收到 n 个线性无关的编码分组时，它可以通过求解一个 n 阶线性方程解码恢复源消息向量。将源 S 所发送的编码分组表示为 $\boldsymbol{X}=[\vec{x}_1,\vec{x}_2,\cdots,\vec{x}_n]^{\mathrm{T}}$，一个 $n\times(m+n)$ 矩阵，第 i 行是 $\vec{x}_i=(x_{i1},x_{i2},\cdots,x_{i(m+n)})\ (1\le i\le n)$。

源 S 形成的 n 个线性无关消息向量，张成空间 $?_X$，因为 $?_X$ 是线性运算闭合的，向量 $\{\vec{x}_1,\vec{x}_2,\cdots,\vec{x}_n\}$ 的线性组合仍然属于 $?_X$，所以网络中传输的编码分组都属于空间 $?_X$。根据定理 1，$?_X$ 的正交空间的维度为 m。将 $?_X$ 的正交空间表示为 $?_X^{\perp}$，由所有满足 $\vec{u}\boldsymbol{X}^{\mathrm{T}}=0$ 的向量 $\vec{u}$ 构成。$?_X^{\perp}$ 由基向量矩阵 $\boldsymbol{U}=[\vec{u}_1,\vec{u}_2,\cdots,\vec{u}_m]^{\mathrm{T}}$ 张成。U 可以表示为 $m\times(m+n)$ 矩阵，第 j 行是 $\vec{u}_j=(u_{j1},u_{j2},\cdots,u_{j(m+n)})\ (1\le j\le m)$，并且 $\boldsymbol{U}\boldsymbol{X}^{\mathrm{T}}=0$。

定理 1　$?_X$ 的正交空间的维度为 m。

证明　从 rank-nullity 定理[13]得知，对于所有的 $m\times n$ 矩阵 $\boldsymbol{A}$，有 $rank(\boldsymbol{A})+nullity(\boldsymbol{A})=n$，其中 $\boldsymbol{A}$ 的正交空间的维度被称为 $\boldsymbol{A}$ 的空度 $nullity(\boldsymbol{A})$。

$\boldsymbol{X}$ 是一个 $n\times(m+n)$ 矩阵，其第 i 行为 $\vec{x}_i$ 以及 $rank(\boldsymbol{X})=n$。运用 rank-nullity 定理，有

$$n+nullity(\boldsymbol{X})=m+n$$

所以正交空间 $?_X$ 的维度为 m。定理 1 得证。

2.2　攻击模型

假设源节点是可信的，假设中间节点和目的节点都可能是恶意的，因为它们可能是攻击者伪装的合法节点或者已经被控制了的合法节点。假设源节点拥有一个多项式运行时间的随机数产生器，而攻击者没有攻陷这个随机数产生器，无法获取源节点生成和保存的随机数。假设攻击者控制了多个恶意节点，可以直接向源节点和目的节点间的传输信道注入污染数据分组，也可以通过恶意节点篡改网络中的合法编码数据分组。一个编码数据分组 $\vec{w}$ 是污染数据分组，当且仅当编码数据分组 $\vec{w}$ 不属于空间 $?_X$。

定义 1　成功的污染攻击。如果攻击者成功构建出一个污染数据分组 $\vec{w}'$，$\vec{w}'$ 能通过安全机制的验证，并且 $\vec{w}'$ 不属于空间 $?_X$。那就说攻击者发起了一次成功的污染攻击。

本文的系统性模型没有限制中间节点的路由算法，也没有规定具体的编码计算方式。这个系统模型适合所有网络编码系统，包括文献[13~15]中所提的编码系统。假设网络已经实现时间同步，每个节点都知道自己与源节点间最大的时钟差异，表示为 D。之前的研究已经提出很多能安全高效地实现网络节点时间同步的协议和机制，例如文献[16,17]。也假设存在端到端消息认证机制，例如传统的数据签名技术和消息认证码技术使每个接收节点都能高效验证解码后原始数据的完整性。

本文关注污染攻击，因为它是网络系统的一个普遍存在而且比较严重的安全威胁。不考虑针对物理层或 MAC 层的攻击，不考虑分组丢失攻击，也不考虑针对特定类型网络编码系统的攻击，例如针对中间节点路由选择的攻击等。

3　ASNC 安全机制

从上一节的系统模型可以得知，网络中传输的合法消息向量拥有一个共同属性，它们都同属于同一线性空间 $?_X$，$?_X$ 中的所有向量都与正交空间 $?_X^{\perp}$ 中的向量正交。数据向量和正交向量由源节点分别发出，中间节点可以通过验证数据向量是否正交于正交向量，从而验证数据向量的合法性。上述正交性原则有效利用了网络编码的本质，衍生出很多的安全机制[12,15,18]。然而，其仍然有 2 个未能解决的问题。

1) 正交性原则的满足并不代表接收数据分组的绝对安全，如果恶意节点能够获取了邻居节点收集的正交向量值，就能很轻易地计算出满足正交性原则的污染数据分组。

2) 接收数据分组的完整性将会在每个中间节点上都进行检查，甚至在没有攻击的情况下也是如此。这就导致了不必要的计算和传输延迟。

为了解决以上问题，提出 ASNC 机制，一个自适应的网络编码安全传输机制，其使中间节点能高效检测出攻击节点产生出的污染数据分组。

3.1　机制描述

ASNC 机制中，一些随机参数被引入到正交向量的生成过程，这些新的验证向量称为 ASNC 验证

向量。下面将逐步描述ASNC安全机制。首先，描述ASNC的参数定义，然后描述源节点启动以及中间节点数据分组验证，最后，描述目的节点数据分组接收。

3.1.1 参数定义

本小节将描述ASNC机制的一些运行参数的定义。把源节点准备分发的数据表示为 X，一个 $n\times(m+n)$ 矩阵，其第 i 行是有限域 F_q 的一个向量 $\vec{x}_i(1\le i\le n)$。此外，$\{\vec{x}_1,\vec{x}_2,\cdots,\vec{x}_n\}$ 张成一个线性空间 $?_X$，其正交空间 $?_X^{\perp}$ 由正交向量矩阵 $\boldsymbol{U}=[\vec{\boldsymbol{u}}_1,\vec{\boldsymbol{u}}_2,\cdots,\vec{\boldsymbol{u}}_m]^{\mathrm{T}}$ 张成，$\boldsymbol{U}$ 是一个 $m\times(m+n)$ 矩阵。设 m_0 是一个常数，且 $1\le m_0\le m$。

3.1.2 源节点启动

源节点不需要改变其发送消息向量的步骤，只需按照网络编码的传输模式处理和发送消息向量。而为了检测污染数据分组，源节点需要每间隔一段时间就生成和分发一系列验证向量，称这些验证向量为ASNC验证向量。ASNC验证向量可以使中间节点高效验证所收到的编码消息向量的合法性。为了让ASNC验证向量具有权威性，源节点用数字签名方法[19]保护ASNC验证向量。源节点启动的具体细节如下所示。

1) 源节点求出源消息向量矩阵 $\boldsymbol{X}$ 的正交向量矩阵 $\boldsymbol{U}$，$\boldsymbol{U}$ 是一个 $m\times(m+n)$ 矩阵且满足 $\boldsymbol{U}\boldsymbol{X}^{\mathrm{T}}=0$，接着源节点从 $\boldsymbol{U}$ 中选出 m_0 行，组成一个 $m_0\times(m+n)$ 的正交向量矩阵 $\boldsymbol{U}_0=[\vec{\boldsymbol{u}}_1,\vec{\boldsymbol{u}}_2,\cdots,\vec{\boldsymbol{u}}_{m_0}]^{\mathrm{T}}$，其中，$\vec{\boldsymbol{u}}_j\boldsymbol{X}^{\mathrm{T}}=0(1\le j\le m_0)$，且 $\{\vec{\boldsymbol{u}}_1,\vec{\boldsymbol{u}}_2,\cdots,\vec{\boldsymbol{u}}_{m_0}\}$ 线性无关，$\vec{\boldsymbol{u}}_j=(u_{j1},u_{j2},\cdots,u_{j(m+n)})(1\le j\le m_0)$。

2) 源节点在 F_q 中生成一个非零随机矩阵，$B=\{b_{jk}\}(1\le j\le m_0,1\le k\le m+n)$。所有 b_{jk} 都是 F_q 中的随机非零元素，其相应的乘法逆元构成逆矩阵 $\boldsymbol{B}^{-1}=\{b_{jk}^{-1}\},b_{jk}b_{jk}^{-1}=1(1\le j\le m_0,1\le k\le m+n)$。在 F_q 选取一个随机数 h 作为验证参数，$\boldsymbol{B}$、$\boldsymbol{B}^{-1}$ 以及 h 都只由源节点保存。

3) 源节点计算2个构成验证向量的验证矩阵 $A=\{a_{jk}\}$，$a_{jk}=h^{b_{jk}}(1\le j\le m_0,\ 1\le k\le m+n)$，计算 $D=\{d_{jk}\}$，$d_{jk}=u_{jk}\cdot b_{jk}^{-1}(1\le j\le m_0,\ 1\le k\le m+n)$。源节点生成ASNC验证向量，$K_{\mathrm{dadp}}=(A,D,t_a)$，$t_a$ 是 K_{dadp} 在源节点建立时的时间戳。源节点对 K_{dadp} 进行数字签名，然后广播 K_{dadp}。

3.1.3 中间节点数据分组验证

中间节点的数据分组验证过程可以细分为4个步骤，分别为初始化、接收、验证以及发送。

1) 初始化

每个编码数据分组包含一个参数 g，其记录了数据分组从上一次验证后经历的跳数。从源节点分发的数据分组，相应的参数 g 为0。

每个中间节点 v 包含一个变量 T_v，其代表了该节点的工作模式。在系统初始化阶段，所有中间节点开始于 $T_v=0$ 模式。

2) 接收

将编码数据分组 $\vec{w}$ 的缓存格式表示为 $(g_w,flag_w,t_w,\vec{\boldsymbol{w}})$，$flag_w$ 是数据分组 $\vec{w}$ 的验证状态，t_w 是数据分组 $\vec{w}$ 的接收时间。当接收到一个新的线性无关编码数据分组 $\vec{w}$，如果 $T_v>0$ 或者数据分组自从上一次数据验证之后已经传输了 l 跳（即 $g_w>l$，其中 l 是一个预设参数），节点 v 把 $flag_w$ 设为1；否则，$flag_w$ 设为2。

3) 验证

当中间节点接收到一个ASNC验证向量，该节点首先验证这个ASNC验证向量是否由源节点签名发出。如果这个ASNC验证向量是可信的，该节点继续向其邻居节点广播该ASNC验证向量。

该节点使用ASNC验证向量 $K_{\mathrm{dadp}}=(A,D,t_a)$ 验证那些接收时间比该ASNC验证向量创建时间更早的（$t_w\le t_a-D$）而且未被验证的数据分组（$flag_w=1$ 或 $flag_w=2$）。D 是网络中节点与源节点的最大时钟差异。节点通过验证下面等式成立与否验证数据分组 $\vec{w}=(w_1,w_2,\cdots,w_{m+n})$ 的正确性

$$[v_{jj}]=\boldsymbol{E}_{m_0} \tag{3}$$

其中，$\boldsymbol{E}_{m_0}$ 是 m_0 阶单位矩阵，矩阵 $\boldsymbol{V}_{m_0}=[v_{jj}]$ 中的元素 $v_{jj}=\prod_{k=1}^{m+n}a_{jk}^{d_{jk}\cdot w_k}\ (1\le j\le m_0)$。如果要求式(3)成立，也就是要求下面的等式成立，即

$$\prod_{k=1}^{m+n}a_{jk}^{d_{jk}\cdot w_k}=1\ \ (1\le j\le m_0) \tag{4}$$

定理2 式(4)对于源节点发送的任意合法数据分组都成立。

证明 对于正交向量矩阵 $\boldsymbol{U}_0=[\vec{\boldsymbol{u}}_1,\vec{\boldsymbol{u}}_2,\cdots,\vec{\boldsymbol{u}}_{m_0}]^{\mathrm{T}}$ 以及合法数据分组 $\vec{w}$，有 $\vec{\boldsymbol{u}}_j\vec{\boldsymbol{w}}^{\mathrm{T}}=0(1\le j\le m_0)$。矩阵 $\boldsymbol{U}$ 的第 j 行可以表示为 $\vec{\boldsymbol{u}}_j=(u_{j1},u_{j2},\cdots,u_{j(m+n)})$。

将合法数据分组 $\vec{w}$ 表示为 $\vec{w}=(w_1,w_2,\cdots,w_{m+n})$，且 $a_{jk}=h^{b_{jk}}$，$d_{jk}=u_{jk}b_{jk}^{-1}$ $(1\le j\le m_0,\ 1\le k\le m+n)$。可以得到以下运算：

$$\begin{aligned}\prod_{k=1}^{m+n}a_{jk}^{d_{jk}w_k}&=\prod_{k=1}^{m+n}\left(h^{b_{jk}}\right)^{\left(u_{jk}b_{jk}^{-1}\right)w_k}\\&=\prod_{k=1}^{m+n}h^{u_{jk}b_{jk}b_{jk}^{-1}w_k}\\&=\prod_{k=1}^{m+n}h^{u_{jk}w_k}\\&=h^{\sum_{k=1}^{m+n}\left(u_{jk}w_k\right)}\\&=h^{\vec{u}_j\vec{w}^{\mathrm{T}}}\\&=h^0\\&=1\end{aligned}$$

所以式(4)对于源节点发送的任意合法数据分组都成立。定理 2 得证。

式(4)是一个简单的线性运算，能够使中间节点高效准确地进行数据分组验证。通过验证的数据分组，$flag_w$ 设为 3，参数 g 设为 0。没有通过验证的数据分组将被丢弃。如果所有的数据分组都通过了验证而 T_v 不为 0，那么 T_v 将被减 1。如果有数据分组被验证为污染数据分组，T_v 将增加 $g\left(1-\frac{g_{\min}}{l}\right)$，$g_{\min}$ 是验证失败的数据分组中最小的参数 g，g 是一个预设参数。为了防止数据溢出，将 T_v 的上限设为 $T_{\max}$，$T_{\max}$ 也是一个预设的参数。

4) 发送

每个节点计算可用数据分组($flag_w=2$ 或者 $flag_w=3$)的线性组合，形成新的编码分组，并且转发该编码分组。使用 $g_{\max}$ 表示所有参与新编码分组组合的可用数据分组的最大参数 g，而新编码分组的参数 g 被设定为 $g_{\max}+1$。

中间节点数据分组验证算法的伪码如算法 1 所示。

3.1.4　目的节点数据分组接收

目的节点也需要用算法 1 验证收到的数据分组，当一个目的节点接收足够的通过验证的线性无关编码数据分组，它可以通过解码这些编码分组恢复原始数据。在往传输协议的上层传送数据前，它使用端到端消息认证机制验证原始数据，这是为了杜绝那些在极小概率下通过ASNC数据分组验证的污染数据分组，极少量的污染数据也能引起上层协议的错误。

算法 1　ASNC 数据分组验证算法

接收数据分组 $\vec{w}$

1) if ($T_v>0$ or $g_w\ge l$) then $flag_w=1$

2) else $flag_w=2$

$K_{\mathrm{dadp}}=(A,D,t_a)$ 验证：

1) Verify all packet $\vec{w}$ with

($flag_w=1$ or $flag_w=2$) and ($t_w\le t_a-D$)

2) Set($g_{w'}=0$ and $flag_{w'}=3$)for all verified packets $\vec{w}'$

3) if there exist is any authentication failure then

4) Set $g_{\min}$ as the minimum g in all packets

that failed authentication

5) if $T_v>T_{\max}$ then $T_v=T_{\max}$

6) else $T_v=T_v+g\left(1-\frac{g_{\min}}{l}\right)$

7)else if $T_v>1$ then $T_v=T_v-1$

8) else $T_v=0$

发送数据分组 $\vec{w}''$：

1) Select a subset of packets that

($flag_w=2$ or $flag_w=3$) to form a coded packet $\vec{w}''$

2) Set $g_{\max}$ to be the maximum g in the selected packets

3) Set $g_{w''}=g_{\max}+1$

在 $T_v>0$ 模式中，中间节点延缓编码数据分组传输，直到它们可以通过后续的 ASNC 验证向量验证。在 T_V=0 模式中，中间节点没有等待 ASNC 验证向量的验证，直接编码和转发接收到的数据分组，除非该数据分组自从上一次验证开始已经传输了至少 l 跳。未被验证的数据分组传输范围的限制使得污染数据分组最大污染范围被限制为 l 跳。一个节点停留在 $T_V>0$ 模式的时间与相距攻击节点的距离是成反比的，所以与攻击节点距离近的节点趋向于数据分组验证，而与攻击节点距离远的节点趋向于不产生延迟直接转发数据分组。通过这种方式，本文的方案可以动态地调整中间节点的验证模式，可以很好地自适应于当前的网络安全态势。

应该注意编码数据分组的参数 g 不需要安全保护。如果 g 被篡改增大，相应的污染数据分组在传播更少跳数之后就必须经历 ASNC 数据分组验证。

如果将被限制传播更小的范围，如果 g 被篡改减小，攻击节点的邻居检测出污染数据分组后，它们的 T_v 将相应增加更多，从而邻居节点停留在 $T_v>0$ 模式的时间更加长，可以更长时间地隔绝污染节点的污染数据分组。ASNC 验证向量也不需要可靠传输，如果一个节点不能接收到当前版本的 ASNC 验证向量，它可以使用源节点在下一时间间隔发送的 ASNC 验证向量来验证缓存里的未验证数据分组。

3.2 批量验证

ASNC 数据分组验证可以扩展为对一组编码数据分组高效地批量验证。假设 $W=\{\vec{w}_1,\vec{w}_2,\cdots,\vec{w}_l\}$ 是一个未验证编码数据分组集合，并且里面的编码数据分组都在当前版本验证数据分组生成之前被节点 v 接收到。v 首先计算集合 W 元素的一个随机线形组合，也就是，$\vec{w}=\sum_{i=1}^{l}j_i\vec{w}_i$，其中，$j_i(1\le i\le l)$ 是 F_q 中的非零随机元素。v 使用上文的 ASNC 数据分组验证方法验证组合数据分组 $\vec{w}$。批量验证的漏检率可以通过使用不同的随机系数重复上述批量验证来降低。

如果 W 通过了批量验证，W 里面所有的编码数据分组都可以被认为是合法的。否则，W 里面污染的数据分组可以使用例如二叉树检索的方法定位出来。

4 安全性分析

之前提到，ASNC 验证向量使用数字签名保护，因此攻击者不能向网络注入伪造的 ASNC 验证向量。对于攻击者来说，唯一合理的攻击选择是产生污染的数据分组，并使之通过 ASNC 数据分组验证。

引理 1 假设有限域 F_q 中 m_0 个线性独立的 $n+m$ 维向量组成了一个 $m_0\times(m+n)$ 矩阵 $\boldsymbol{U}_0$，那么一个随机的 $n+m$ 维向量与 $\boldsymbol{U}_0$ 正交的概率为 $\frac{1}{q^{m_0}}$。

证明 所有与 $\boldsymbol{U}_0$ 正交的 $n+m$ 维向量，都属于 $\boldsymbol{U}_0$ 的正交空间 $?_{U_0}^{\perp}$。借鉴定理 1 的分析，可以得知 $rank(\boldsymbol{U}_0)+nullity(\boldsymbol{U}_0)=n+m$，可以得出 $\boldsymbol{U}_0$ 的正交空间的维度为 $n+m-m_0$。一个随机的 $n+m$ 维向量 $\vec{w}$ 属于 $\boldsymbol{U}_0$ 的正交空间 $?_{U_0}^{\perp}$ 里的向量的概率为

$$Pr(\vec{w}\in ?_{U}^{\perp})=\frac{q^{n+m-m_0}}{q^{n+m}}=\frac{1}{q^{m_0}}$$

所以一个随机的 $n+m$ 维向量与 $\boldsymbol{U}_0$ 正交的概率为 $\frac{1}{q^{m_0}}$。引理 1 得证。

定义 2 编码数据分组能被 ASNC 验证向量安全验证。如果编码数据分组 $\vec{w}$ 的接收时间比 ASNC 验证向量 K_{dadp} 的构建时间更早，那么 $\vec{w}$ 能被 K_{dadp} 安全验证。

在 ASNC 安全机制中，ASNC 验证向量 K_{dadp} 的构建过程加入了一系列非零随机元素，攻击者希望从历史版本 ASNC 验证向量或者编码数据分组得到新版本 ASNC 验证向量任何有价值的构建信息是计算不可行的。同时，中间节点仅使用 K_{dadp} 验证那些接收时间比其构建时间早的编码数据分组，就表示如果 $\vec{w}$ 是污染数据分组，攻击者构建 $\vec{w}$ 时 K_{dadp} 还没构建，攻击者不可能截获到 K_{dadp} 用以分析且优化其攻击手段，中间节点也就可以利用 K_{dadp} 安全验证 $\vec{w}$ 的合法性。攻击者在没有任何 ASNC 验证向量有价值构建信息的前提下，最优的攻击模式只能是发送随机污染数据分组希望能通过 K_{dadp} 的验证。

定理 3 攻击者发送的随机 $n+m$ 维污染数据分组成功通过 ASNC 验证的概率不大于 $\frac{1}{q^{m_0}}$。

证明 源节点发送的编码消息空间为 $?_X$，其正交空间 $?_X^{\perp}$ 由正交向量矩阵 $\boldsymbol{U}_0=[\vec{\boldsymbol{u}}_1,\vec{\boldsymbol{u}}_2,\cdots,\vec{\boldsymbol{u}}_{m_0}]^{\mathrm{T}}$ 张成，$\boldsymbol{U}_0$ 是一个 $m_0\times(m+n)$ 矩阵。结合定理 2 的分析过程，随机污染数据分组 $\vec{w}$ 能通过 ASNC 验证必须符合 $\vec{\boldsymbol{u}}_j\vec{\boldsymbol{w}}^{\mathrm{T}}=0(1\le j\le m_0)$，也就是 $\boldsymbol{U}_0\vec{\boldsymbol{w}}^{\mathrm{T}}=0$。这就要求 $\vec{w}$ 与 $\boldsymbol{U}_0$ 正交。

根据引理 1，得知一个随机 $n+m$ 维向量与 $\boldsymbol{U}_0$ 正交概率为 $\frac{1}{q^{m_0}}$，所以一个随机 $n+m$ 维污染数据分组成功通过 ASNC 验证的概率不大于 $\frac{1}{q^{m_0}}$。定理 3 得证。

定理 4 使 $K_{\text{dadp}}=(\boldsymbol{A},\boldsymbol{D},t_a)$ 为节点 v 中当前版本的 ASNC 验证向量，使 $W=\{\vec{w}_1,\vec{w}_2,\cdots,\vec{w}_l\}$ 为一个分组含污染数据分组的集合。W 成功通过 y 次不同随机系数的批量验证的概率为不大于 $\frac{1}{q^{y}}+\frac{1}{q^{m_0}}$。

证明 在这 y 次批量验证中的第 j $(1\le j\le y)$ 次批量验证中，假设提交验证的组合数据分组为 $\vec{w}_j=\sum_{i=1}^{l}j_i\vec{w}_i$，其中，$j_i(1\le i\le l)$ 是 F_q 中的非零随机元素。合法数据空间 $?_X$ 是一个 n 维线性空间，

编码数据分组的前 n 项元素代表全局编码向量。假设空间 $?_X$ 中与数据分组集合 $W=\{\vec{w}_1,\vec{w}_2,\cdots,\vec{w}_l\}$ 中各数据分组全局编码向量相同的合法数据为 $E'=\{\vec{w}'_1,\vec{w}'_2,\cdots,\vec{w}'_l\}$，这样的合法数据分组集合可以在空间 $?_X$ 中唯一确定。使用同样的随机系数计算合法组合数据分组 $\vec{w}'_j=\sum_{i=1}^{l}j_i\vec{w}'_i$。

假设事件 Eq_j 代表在第 j 次批量验证中 $\vec{w}_j=\vec{w}'_j$，事件 Pw_j 代表在第 j 次批量验证中 $\vec{w}_j$ 成功通过 ASNC 验证，而事件 PW_j 代表污染数据分组集合 W 成功通过第 j 次批量验证。

可以知道，$PW_j=Eq_j\cup(\overline{Eq}_j\cap Pw_j)$。假设 $Py=\bigcap_{j=1}^{y}PW_j$，也就是说事件 Py 代表污染数据分组集合 W 成功通过 y 次不同随机系数的批量验证。

$$\begin{aligned}Pr(Py)&=Pr\left(\bigcap_{j=1}^{y}PW_j\right)\\&=Pr\left(\bigcap_{j=1}^{y}(Eq_j\cup(\overline{Eq}_j\cap Pw_j))\right)\\&=Pr\left(\bigcap_{j=1}^{y}Eq_j\cup\bigcap_{j=1}^{y}(\overline{Eq}_j\cap Pw_j)\right)\\&\le Pr\left(\bigcap_{j=1}^{y}(Eq_j)\right)+Pr\left(\bigcap_{j=1}^{y}(\overline{Eq}_j\cap Pw_j)\right)\\&\le Pr\left(\bigcap_{j=1}^{y}(Eq_j)\right)+Pr(\overline{Eq}_1\cap Pw_1)\\&=\prod_{j=1}^{y}Pr(Eq_j)+Pr(\overline{Eq}_1\cap Pw_1)\\&\le\prod_{j=1}^{y}Pr(Eq_j)+Pr(Pw_1)\end{aligned}\tag{5}$$

上面分析中，第三行等号成立的原因是因为事件 Eq_j 和事件 $\overline{Eq}_j\cap Pw_j$ 不相交，倒数第二行等号成立的原因是因为事件 $Eq_j(1\le j\le y)$ 相互独立，在每次的批量验证中形成组合数据分组的随机系数都是独立选取的。

由定理 3 可知，$Pr(Eq_j)$ 代表在第 j 次批量验证中 $\vec{w}_j=\vec{w}'_j$ 的概率，即 $\sum_{i=1}^{l}j_i\vec{w}_i-\sum_{i=1}^{l}j_i\vec{w}'_i=0$ 的概率。网络中的编码数据分组都是 $n+m$ 维向量，假设 $\vec{w}_i=(w_{i1},w_{i2},\cdots,w_{i(m+n)})(1\le i\le l)$，因为 $\vec{w}'_i$ 前 n 项元素与 $\vec{w}_i$ 相同，设 $\vec{w}'_i=(w_{i1},\cdots,w_{in},w'_{i(n+1)},\cdots,w'_{i(m+n)})$。因为 $E=\{\vec{w}_1,\vec{w}_2,\cdots,\vec{w}_l\}$ 是一个包含污染数据分组的集合，必定存在某一个 $i\ (1\le i\le l)$ 满足 $\vec{w}_i\ne\vec{w}'_i$，也就是必定存在某个 $k\ (n+1\le k\le n+m)$ 满足 $w_{ik}\ne w'_{ik}$，不失一般性，假设 $w_{1(n+1)}\ne w'_{1(n+1)}$。同时，设 $z_{ik}=w_{ik}-w'_{ik}(1\le i\le l,n+1\le k\le n+m)$，由前面假设知 $w_{1(n+1)}\ne w'_{1(n+1)}$，所以 $z_{i(n+1)}\ne 0$，所以 $z_{i(n+1)}$ 在 F_q 中存在唯一的乘法逆元，记为 $z^{-1}_{i(n+1)}$。

事件 Eq_j 代表 $\vec{w}_j=\vec{w}'_j$，其发生概率等于下面等式成立的概率 $\sum_{i=1}^{l}j_i(\vec{w}_i-\vec{w}'_i)=0$，也就是对于所有的 $n+1\le k\le n+m$，$\sum_{i=1}^{l}j_i(w_{ik}-w'_{ik})=0$ 都成立。假设事件 $Eq_{j(n+1)}$ 代表 $\sum_{i=1}^{l}j_i(w_{i(n+1)}-w'_{i(n+1)})=0$ 成立，显而易见 $Eq_j\subseteq Eq_{j(n+1)}$。

对于事件 $Eq_{j(n+1)}$，因为 $z_{i(n+1)}=w_{i(n+1)}-w'_{i(n+1)}$，所以 $\sum_{i=1}^{l}j_i(w_{i(n+1)}-w'_{i(n+1)})=0$ 可以转化为

$$j_1z_{1(n+1)}+j_2z_{2(n+1)}+\cdots+j_lz_{l(n+1)}=0\tag{6}$$

前面已经假设 $z^{-1}_{i(n+1)}$ 是 $z_{i(n+1)}$ 在 F_q 中唯一的乘法逆元，式(6)可以变换为

$$j_1=-z^{-1}_{i(n+1)}(j_2z_{2(n+1)}+\cdots+j_lz_{l(n+1)})\tag{7}$$

确定的 $\{j_2,\cdots,j_l\}$、$\{z_{1(n+1)},z_{2(n+1)},\cdots,z_{l(n+1)}\}$ 对应着一个确定的 j_1 能让式(7)成立。在 ASNC 机制中，中间节点在收到数据分组集合 $E=\{\vec{w}_1,\vec{w}_2,\cdots,\vec{w}_l\}$ 后才确定批量验证系数 $\{j_1,\cdots,j_l\}$，所以就算攻击者希望通过调整 $\{z_{1(n+1)},z_{2(n+1)},\cdots,z_{l(n+1)}\}$ 的值来让式(7)成立，其成功的概率也就等同于从有限域 F_q 中随机选择一个元素希望能与 j_1 相等的概率，这样的概率为 $\frac{1}{q}$。

所以 $Pr(Eq_{j(n+1)})=\frac{1}{q}$，因为 $Eq_j\subseteq Eq_{j(n+1)}$，所以 $Pr(Eq_j)\le Pr(Eq_{j(n+1)})$，也就是 $Pr(Eq_j)\le\frac{1}{q}$。

结合式(5)的分析，可知 $Pr(Py)\le\frac{1}{q^y}+\frac{1}{q^{m_0}}$，也就是 W 成功通过 y 次不同随机系数的批量验证

的概率不大于$\frac{1}{q^{y}}+\frac{1}{q^{m_0}}$，定理4得证。

定理5 污染数据分组能不经中间节点验证的污染跳数最大上限为$l+1$。

证明 网络初始化时，所有的中间节点都被设定为$T_v=0$模式开始，这时的网络安全策略是最宽松的。当编码数据分组经过一个诚实节点时，相应的参数g增加1。因此，在传输经过l个诚实节点之后，污染数据分组w'将拥有$g_{w'}=l$，这个污染数据分组将被检测出来并被丢弃。所以污染数据分组能传输的跳数最大上限为$l+1$。定理5得证。

定理6 设$g_{\min}$是攻击节点发送的污染数据分组中参数g的最小值，离攻击节点i跳$(1\le i\le l-g_{\min}-1)$的节点，将增加$g(1-\frac{g_{\min}+i}{l})$个单位的$T_v>0$模式执行时间。在大于等于攻击节点$l-g_{\min}$跳之外的节点，不会增加$T_v>0$模式执行时间。

证明 在ASNC机制中，参数g在每一跳中逐步增加，节点离攻击节点i跳$(1\le i\le l-g_{\min}-1)$，验证出污染数据分组时，将增加停留在$T_v>0$模式$g(1-\frac{g_{\min}+i}{l})$个单位时间间隔。但是，污染数据分组将会在$l-g_{\min}$之前就被检测和丢弃,所以节点在大于等于攻击节点$l-g_{\min}$跳之外的话，不会增加$T_v>0$模式执行时间。

定理6得证。

由定理5得知，l是编码数据分组必须接受一次验证的额定跳数，g控制了节点验证出污染之后停留在$T_v>0$模式的时间。可以使用一个较小的l和较大的g增加方案的安全性，当然这也会使更多的中间节点处在$T_v>0$验证模式中。使用一个较小的l时，未经验证的数据分组仅被允许传输较少的跳数;而使用一个较大的g时，检测到污染数据分组后将增加较多的$T_v>0$模式停留时间。所以在选择l和g时，应该平衡安全的弹性力度和验证延迟之间的关系。

相对来说，可以忍受网络延迟的网络，例如无线传感网的编码升级以及无线网状网的大文件传输，可以使用较小的l和较大的$?$去增加安全的弹性力度。对网络延迟很敏感的网络，例如声音和视频播放，可以使用较大的l和较小的$?$来减小验证停留。对于攻击比较少发生的网络，可以使用较大的l去减小延迟，而用较大的$?$去限制攻击发生时污染数据分组的传播。

5 复杂度分析

ASNC机制的主要包含2个算法，一个是源节点启动算法，一个是中间节点验证算法。下面将分析这2个算法的复杂度。假设$MultCost(q)$是在F_q上做一次乘法运算所需的运算量，对于$1\le z\le m$，在F_q上计算所有的Y^z，平均需要$\frac{m}{2}$次乘法运算。假设源节点要发送的一代数据大小为M，根据网络编码的传输模型把数据切分为n块，每块都是由m个有限域F_q元素组成的向量。每个数据块的大小为$L=m\mathrm{lb}(q)$，所以可以得知$M=mn\mathrm{lb}(q)$。

如3.1.2节所示，源节点的启动包含3个步骤。步骤1）需利用高斯消元法求解$m+n$阶线性方程组，其时间复杂度为$O\left((m+n)^3\right)$。然而该部分运算只在源节点进行新一代数据发送前进行，对于相同的源消息向量$\boldsymbol{X}$，该部分计算只进行一次。步骤2）中源节点生成随机矩阵$\boldsymbol{B}$和$\boldsymbol{B}^{-1}$，所需时间取决于随机数产生器的效率。在实际应用过程中，源节点可以预先计算生成若干组符合条件的矩阵$\boldsymbol{B}$和$\boldsymbol{B}^{-1}$，需要时随机选取一组使用。这部分的时间复杂度可以认为是$O(1)$。步骤3）中源节点生成并分发ASNC验证向量$K_{\mathrm{dadp}}=(\boldsymbol{A},\boldsymbol{D},t_a)$，需要计算验证矩阵$\boldsymbol{A}$和$\boldsymbol{D}$，$\boldsymbol{A}$的计算需要$\frac{\mathrm{lb}(q)}{2}m_0(m+n)\times MultCost(q)$的计算量，$\boldsymbol{D}$的运算需要$m_0(m+n)\times MultCost(q)$。不过，如果同一代的数据采用相同的随机矩阵$\boldsymbol{B}$和$\boldsymbol{B}^{-1}$，那么$\boldsymbol{A}$的计算在一代数据中也只需进行一次。

源节点启动最耗时是步骤1)，但是步骤1)只在源节点发送新一代数据时需要进行，其运行次数与源数据切分成代的数量相等。步骤2)可以在常数时间完成，而步骤3)的运行次数跟ASNC验证向量的发送间隔有关。

中间节点的验证算法复杂度，主要就是式(4)的复杂度，为$\frac{\mathrm{lb}^2(q)}{2}m_0(m+n)\times MultCost(q)$。中间节点的需要运行的验证次数与所接收到的消息向量个数有关。

可以看出，由于ASNC机制有效利用了网络编码的正交性原则，其时间复杂性基本都是若干乘法运算。ASNC机制不需要额外的安全通道，也没有在消息向量中添加附加的安全签名信息。ASNC机

制只需增加发送若干 ASNC 验证向量，每个 ASNC 验证向量的大小是 $2(m+n)\mathrm{lb}(q)$，如果同一代的数据采用相同的随机矩阵 $\boldsymbol{B}$ 和 $\boldsymbol{B}^{-1}$，那么除了首个发送间隔之外源节点都不用计算和发送验证矩阵 $\boldsymbol{A}$ 的信息，这时 ASNC 验证向量的格式为 $K_{\mathrm{dadp}}=(\boldsymbol{D},t_a)$，其数据大小是 $(m+n)\mathrm{lb}(q)$。

ASNC 验证向量由数字签名方法保护，数字签名提供了高度的安全性，但是它具有一定的计算开销。然而，在 ASNC 机制中，数字签名只用来保护 ASNC 验证向量，源节点在每个 ASNC 验证向量分发间隔中向网络分发 $|GO(S)|$ 个 ASNC 验证向量到网络中。在每个分发间隔，每个节点 v 只用验证 1 次 ASNC 验证向量的签名。在本文的安全方案中，源节点分发编码数据的方式并没有改变，同时消息向量中并没有加入附加的安全信息，在消息向量中加入附加信息可能会大量消耗带宽。另一方面，基于同态的协同安全机制需要中间节点进行复杂的同态运算，但本文的安全机制仅需简单的矩阵乘法运算，这使 ASNC 机制的数据分组验证过程快速和轻便。

接下来讨论一些已有的抗污染攻击安全方法的计算量。使用同态散列方法可以让中间节点根据消息向量的同态散列签名值检测出污染数据，但是就如文献[13]所介绍的，同态散列方法的签名过程比传统的 SHA1 散列算法计算量还要大。中间节点每次进行同态散列验证，其计算复杂度为 $(\frac{L}{2}l_q+n)\times MultCost(p)$，其中，$L$ 是数据块的大小，l_q 是一个用作安全参数的大素数，p 也是一个大素数，$MultCost(p)$ 是在 Z_p 上做一次乘法运算所需的运算量。因为同态散列方法需要的运算量比较大，所以有研究者采用一定概率进行同态散列检测以减低同态散列方法的计算复杂度，例如协同同态散列机制[20]。

Null keys 机制[12]中，源节点直接给中间节点发送正交向量，可以实现快速的污染检测。但是其安全性不高，不能抵抗恶意节点的合谋污染攻击。Null keys 机制的启动过程同样需要通过高斯消元法求解 $m+n$ 阶线性方程组，其时间复杂度为 $O((m+n)^3)$。而其验证过程只需判断收到的信息是否与正交向量正交，验证过程最多需要 $m(m+n)\times MultCost(q)$ 的计算量。

6　实验结果

本节将通过模拟实验验证 ASNC 安全机制的安全效率。同时，将 ASNC 安全机制与 CHRISTOS 等人提出的协同同态散列机制[20]和 ELIAS 等提出的 Null keys 机制[12]相比较。网络实验拓扑是一个有向图，其拥有一个源节点。恶意节点的攻击模式如第 2 节介绍。实验模拟环境中，每个单位时间节点都可以从入度链路接收数据分组以及从出度链路发送数据分组，同样每个单位时间恶意节点可以向其每条出度链路发送一个随机污染数据分组。本章的实验结果都是多次运行结果的平均值。实验环境包含 100 个节点，每 2 个节点之间直接链路相连的概率为 p。假设链路中节点的本地时钟都已经同步，而且最大的时钟差异为 $D=100$ ms。网络编码模型中的有限域大小为 $q=257$，网络编码的每一代大小为 $n=32$，每个数据块大小为 $m=128$，$m_0=2$。源节点每发送 32 个编码数据分组就计算和分发一次 ASNC 验证向量。ASNC 机制中的参数使用 $l=8$ 和 $g=15$，这样的参数设置符合小规模网络的安全性和传输效率要求。

图 1 展示了网络编码系统面临污染攻击的安全态势。使用恶意节点百分比衡量攻击力度，使用被污染节点百分比衡量被污染的程度。图中的 p 表示网络中节点之间直接链路相连的概率。从图 1 中可以看出，在 3 种概率 p 的情形下，恶意节点百分比的增加都会导致污染节点百分比明显增加，说明恶意节点的数量越多，污染力度越大，中间节点也面临的污染威胁也越大。同时，随着节点相连概率 p 的增大，恶意节点的污染能力进一步增强。节点相连概率高代表了网络中链路状态好，这时恶意节点可以轻松将大量的污染数据分组注入网络中，从而让网络的污染程度进一步恶化。可以看到，在面临大规模恶意节点的污染攻击时，网络的安全态势不容乐观。

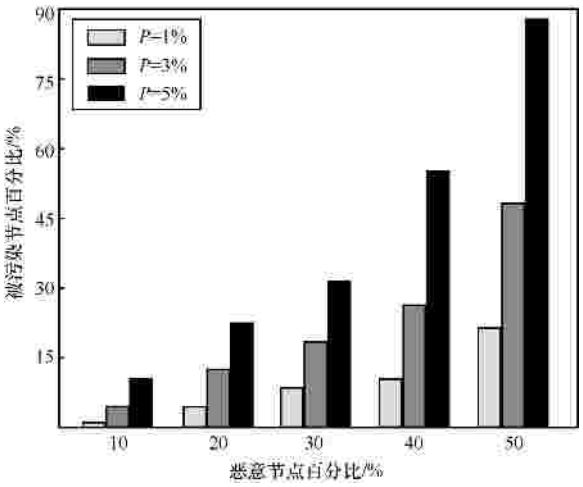

图 1　网络编码面临污染攻击的安全态势

在协同同态散列机制中，节点使用同态散列方法验证接收数据块的合法性。但是为了减轻计算效率的负担，中间节点对接收到的数据分组实施同态验证操作的概率为 pc。当发现污染数据分组时，中间节点向该污染的来源节点和污染的扩散节点都发送警报信息。在 Null keys 机制中，中间节点利用源数据的正交空间来验证接收的数据分组。

图 2 展示了网络中存在 5 个恶意节点时，ASNC 机制和其他安全机制之间的网络数据流累积分布函数对比。可以看到 ASNC 机制比其他机制运行效率更优。ASNC 机制中，大约 80%的数据流可以达到 90 kbit/s 以上，而 Null keys 机制只有 61%的数据流达到 90 kbit/s 以上。对于验证概率为 30%的协同同态散列机制，没有能达到 90 kbit/s 以上的数据流，其 80 %的数据流都低于 20 kbit/s。这是因为就算执行 30%概率的同态验证，网络的传输性能也大受影响。在 ASNC 机制中，只有恶意节点附近的中间节点才倾向于执行数据分组验证，其能自适应与当前网络安全态势，所以减少了不必要的数据分组验证延迟，这也使 ASNC 机制能优于其他安全机制。

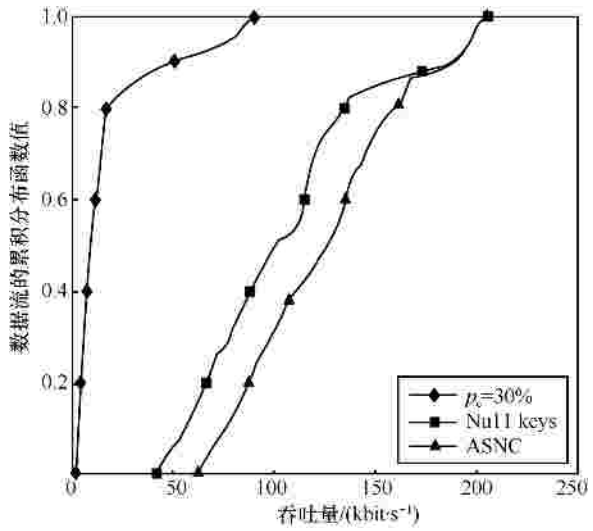

图 2 网络数据流累积分布函数对比

表 1 展示了检测概率为 30%的协同同态散列机制、Null keys 机制以及参数 $m_0=2$ 和 $m_0=3$ 的 ASNC 机制的运算时间对比情况。可以看出，协同同态散列的运算消耗时间比较长，因为同态签名和验证过程都需要进行大量运算操作。而 Null keys 机制和 ASNC 机制的源节点启动时间相差不多，Null keys 机制稍快，无论是 $m_0=2$ 还是 $m_0=3$ 的 ASNC 机制，消息验证过程也还是毫秒级的，而且根据之前的安全性分析，$m_0=2$ 时恶意节点成功发动污染攻击的概率为 $\frac{1}{66\,049}$，$m_0=3$ 时恶意节点成功发动污染攻击的概率为 $\frac{1}{16\,974\,593}$，可以实现非常高的安全性。

表 1 运算时间对比

机制种类	源节点启动时间/ms	消息验证时间/ms
P_c=30%	3 810	1 430
Null keys	67.26	0.185
ASNC($m_0=2$)	69.47	0.378
ASNC($m_0=3$)	69.53	0.508

7 结束语

本文研究了 ASNC 机制，一个能高效抵抗污染攻击的自适应网络编码传输机制，其利用网络编码的时间和空间特性有效控制污染数据分组的传播。在 ASNC 安全机制中，中间节点能利用 ASNC 验证向量安全验证接收到的编码数据分组的合法性，而攻击者不可能获取到任何当前版本的 ASNC 验证向量有价值构建信息。对于 ASNC 机制，攻击者最优的攻击模式只能是通过发送随机污染数据分组希望能通过 ASNC 验证向量的验证。但 ASNC 的验证过程提供高标准的安全保证，随机污染数据分组只有很低的概率能通过验证。同时，还进一步提出 ASNC 机制的批量验证方法，可以进一步提升中间节点一次验证多个待验证数据分组的效率。

ASNC 机制有效利用网络编码的本质特性，不需要额外的安全数据通道和数据分组加密操作。实验模拟现实，ASNC 机制能够有效抵抗污染攻击，与不具有自适应能力的机制相比具有更高的效率。本文的安全机制高效地限制污染和孤立恶意节点。有效控制污染数据分组的传播，可以有助于进一步定位恶意节点的位置，是网络编码系统安全的重要保障。

参考文献：

[1] ADRIAN B, BONDY J A, MURTY U S R. Graph Theory: An Advanced Course[M]. London: Springer London Ltd, 2008.

[2] AHLSWEDE R, CAI N, LI S R. Network information flow[J]. IEEE Transactions on Information Theory, 2000, 46:1204-1216.

[3] SHWE H Y, ADACHI F. Power efficient adaptive network coding in wireless sensor networks[A]. Proceedings of IEEE Inter ional Con-

ference on Communications (ICC)[C]. Kyoto, 2011.1-5.

[4] WU H, ZHENG J. Efficient network coding-based multicast retransmission mechanism for mobile communication networks[J]. IET Communications, 2012, 6(2):187-193.

[5] XU J B, WANG X, ZHAO J. Iswifter: improving chunked network coding for peer-to-peer content distribution[J]. Peer-to-Peer Networking and Applications, 2012, 5(1):30-39.

[6] LI S R, YEUNG RW, CAI N. Linear network coding[J]. IEE Transactions on Information Theory, 2003, 49(2):371-381.

[7] CAI N, CHAN T. Theory of secure network coding[J]. Proceedings of the IEEE, 2011, 99(3):421-437.

[8] CAI N, YEUNG R W. Secure network coding on a wiretap network[J]. IEEE Transactions on Information Theory, 2011, 57(1):424-435.

[9] YEUNG R W, CAI N. Network error correction, part I: basic concepts and upper bounds[J]. Communications in Information and Systems, 2006, 6(1):19-36.

[10] CAI N, YEUNG R W. Network error correction, part II: lower bounds[J]. Communications in Information and Systems, 2006, 6(1):37-54.

[11] SILVA D, KSCHISCHANG F R, KOETTER R. A rank-metric approach to error control in random network coding[A]. Proceedings of IEEE Information Theory Workshop on Information Theory for Wireless Networks[C]. Solstrand, 2007. 1-5.

[12] ELIAS K, BAOCHUN L. Null keys: limiting malicious attacks via null space properties of network coding[A]. Proceedings of IEEE International Conference on Computer Communications (INFOCOM) C]. Anchorage, 2009. 1224-1232.

[13] KROHN M N, FREEDMAN M J, MAZIERES D. On-the-fly verification of rateless erasure codes for efficient content distribution[A]. Proceedings of IEEE Symposium on Security and Privacy[C]. Oakland, 2004. 226-240.

[14] LI Y P, YAO H Y, CHEN M H. RIPPLE authentication for network coding[A]. Proceedings of IEEE International Conference on Computer Communications (INFOCOM)[C]. San Diego,2010. 2258-2266.

[15] JING D, REZA C, CRISTINA N R. Practical defenses against pollution attacks in intra-flow network coding for wireless mesh networks[A]. Proceedings of ACM Conference on Wireless Network Security[C]. Zurich, 2009. 111-122.

[16] ADRIAN P, RAN C, TYGAR J D. The TESLA broadcast authentication protocol[J]. RSA CryptoBytes, 2002, 5:1-13.

[17] KUN S, PENG N. Secure and resilient clock synchronizat on in wireless sensor networks[J]. IEEE Journal on Selected Areas in Communications, 2006, 24(3):395-408.

[18] DAN B, DAVID F, JONATHAN K. Signing a linear subspace: signature schemes for network coding[A]. Proceedings of International Conference on Practice and Theory in Public Key Crypto hy[C]. Orange County, California, 2009.68-87.

[19] CHARLES D, JAIN K, LAUTER K. Signatures for network coding[A]. Proceedings of Annual Conference on the Information Sciences and Systems[C]. Princeton, 2006.857-863.

[20] CHRISTOS G, PABLO R. Cooperative security for network coding file distribution[A]. Proceedings of IEEE Internationa Conference on Computer Communications (INFOCOM)[C]. Barcelona, 2006.1-13.

作者简介：

何明（1984-），男，广东从化人，国防科学技术大学博士生，主要研究方向为网络安全与网络计算、密码学。

邓罡（1983-），男，贵州晴隆人，国防科学技术大学博士生，主要研究方向为数据中心网络管理、云计算。

王宏（1964-），男，湖南益阳人，博士，国防科技大学副研究员，主要研究方向为网络流量分析。

龚正虎（1945-），男，湖南长沙人，国防科技大学博士生导师，主要研究方向为计算机网络与通信。