基于 GID 的车联网数据安全方案

庞立君,廖春伟,黄 波,赵海涛
(南京邮电大学 通信与信息工程学院,江苏 南京 210003)

摘 要:车联网在“端-管-云”三层架构的基础上,提供丰富的智能交通综合服务。然而,将数据放置在云端处理和存储,加大了数据被非法用户窃取的风险。为此,提出了基于网络基因 GID(Gene IDentification)的车联网数据安全方案。利用 GID 标示数据上传者和云中可被外界访问的数据,保证数据在上传和存储在云中时的唯一性。进行数据访问时,通过比较待访问数据的网络基因与预先提取的可访问数据的基因是否一致,如果一致,则允许数据流出云端,供用户使用。经过分析和仿真表明,基于 GID 的车联网数据安全方案在保证数据安全性的同时,可以减少云端存储空间的浪费,增大数据上传的速率。

关键词:车联网;云存储;数据安全;网络基因
中图分类号:TP301 **文献标识码**:A **文章编号**:1673-629X(2016)04-0101-04
doi:10.3969/j.issn.1673-629X.2016.04.022

Data Security Scheme of IOV Based on GID

PANG Li-jun,LIAO Chun-wei,HUANG Bo,ZHAO Hai-tao
(College of Telecommunications & Information Engineering,Nanjing University of Posts and Telecommunications,
Nanjing 210003,China)

Abstract:As a specific application of IOT (Internet Of Things),IOV (Internet Of Vehicle) based on "terminal-pipe-cloud" three-tier system,provides a wealth of intelligent transportation integrated services. However,the data placed in the cloud processing can increase the risk that the illegal user data can steal the secrets. For this,a cloud data security scheme of IOV based on network GID (Gene IDentification) is put forward. Using GID to label data uploaded by data owners and cloud data accessed by the outside world can ensure the uniqueness of the data at the time of uploading and storing in the cloud. When data users want to access data,they can access and use the data only when the gene of data to be accessed is consistent with the pre-extracted gene of data which can be accessed. Analysis and simulation show that the cloud data security scheme of IOV based on network GID not only ensures data security but also reduces the waste of cloud storage space,while increasing the data upload speeds.

Key words:IOV;cloud storage;data security;network gene

0 引 言

作为物联网在车辆领域的具体应用,车联网[1]对通信需求具有即时响应[2]的特点,这些通信需求包括车与车、车与路、车与人、车-路-云计算平台的通信等,并且对信息的采集、感知和处理也十分多样化[3]。车联网[4]通过获取车辆运行参数和道路等交通基础设施使用状况,对道路交通路况进行实时监控和调度,从而为车辆行驶提供丰富的智能交通综合服务。然而,车联网所提供的这些业务的运行都需要“端-管-云”[5](通信、传感、计算终端+数据上传网络+云计算中心)平台提供。其中:“管”将车辆运行状态和相关数据进行实时上报;“端”是车与车、车与路、车与人、人与云交互与展示方式,使用户通过智能终端或 App 进行车联网功能展示和体验,进而实现车联网中人车互动;“云”是整个车联网的核心,实现对海量涉车数据的存储、计算、管理、监控、分析、挖掘及应用,并为其他用户提供数据访问能力。

车云平台具有不分地域,不分业务种类,车辆全网

收稿日期:2014-12-09 **修回日期**:2015-04-12 **网络出版时间**:2016-03-22
基金项目:国家自然科学基金资助项目(61302100,61471203,61201162);中国博士后研究基金(2013M531391);教育部博士点基金(20133223120002);江苏省基础研究计划-重点研究专项基金(BK2011027,BK2012434);江苏省博士后研究基金(1202083C)
作者简介:庞立君(1991-),女,硕士研究生,研究方向为下一代通信网络技术;黄 波,博士,讲师,研究方向为无线网络与泛在通信;赵海涛,博士,博士后,副教授,研究方向为无线网络与泛在通信。
网络出版地址:http://www.cnki.net/kcms/detail/61.1450.TP.20160322.1518.024.html

透视,提供统一集中的开放车联业务平台的特点[4]。越来越多的用户和企业将自己的数据上传至云端进行存储和处理,当多个用户同时向一定区域云发送数据时,会造成云空间紧张,以及数据所有者上传数据的速率降低,甚至因网络阻塞中断数据传输任务[6-7]。将数据放置在云端进行存储,数据所有者也就失去了对数据的直接控制,车辆中如行车轨迹,车辆最近时间进行导航的地点和路线等涉及用户隐私的数据,直接存放在云端,增大了被非法用户窃取的风险。

GID(Gene IDentification)是在物理空间(Physical Space,PS)和信息空间(Cyber Space,CS)中都具有唯一性和统一的可信任性的识别码,涉及信息物理系统 CPS、物联网、互联网等领域,用于解决上述领域中的实体身份识别问题。GID 可实现 CS 和 PS 的映射。GID 在 CS 中是可以全面地描述网络所有终端以及电子智能实体等网络实体自身固有特征标识的系统,并具有全球(移动)识别能力的、能够唯一性地代表使用载体个性的新型 ID 结构特点。由于云存储环境具有多个数据上传者和数据使用者的特点,数据的上传、存储和访问各个环节更易受到外界的攻击。由于 GID 具有全球唯一的特性,因此可以用来唯一辨别用户和数据内容。

由上述可以看出,GID 在车联网云端数据的安全方面具有很重要的意义和价值[8]。针对云空间紧张,数据上传速率降低甚至为 0 和数据被非法用户访问的问题[9],文中提出一种新型的解决车联网环境下数据安全的方案—基于 GID 的车联网数据安全方案。

1 系统方案设计

1.1 设计目标

为实现云中数据安全性的目标,有以下重要的设计原则:

(1)数据上传过程中,系统可以产生唯一标示数据上传用户的 DOGID,以便云端可以准确无误、迅速地识别上传用户。如果用户的数据允许放在此块云区域内,给用户开辟数据存储空间,允许用户上传数据;反之,则立即阻止数据的上传。这样,由于并不是所有的数据用户都可以将自身的数据上传至云端,将会大大加快授权用户数据的上传速率,同时也节约了云中存储单位的浪费。

(2)对于存储在云中并且可以被外界访问的数据,云服务器可以给每条数据内容创建唯一的标识 DGID,并将 DGID 存储在数据基因数据库 DGIDD 中,以便用户访问时进行信息的匹配和比较。

(3)在数据使用者访问数据时,如果用户访问的数据不允许流出网络,系统应该尽快识别和阻止数据的流出。例如,数据真实泄露点与系统检测到泄露点之间的间隔越小越好。

1.2 系统模型

系统假设合法用户可以通过执行预先明确的协议,如 HTTP,FTP 等来访问数据(只有可以通过这些协议的用户才允许进入网络传输数据)。即用户数据访问请求可以被云端接受,并且云端可以完全按照访问请求返回相应的数据,但是,在数据流出系统之前,对用户待访问的数据提取数据网络基因 DGID′,并与 DGIDD 中的允许外界访问数据的网络基因进行比较,如果匹配成功,则将数据返回给访问用户。系统总体架构图如图 1 所示。

系统分为数据上传者(DO)、云端设备(CSP)、数据 GID 数据库(DGIDD)、数据使用者(DU)四部分。各部分详细作用如下:

DO:在车辆网中,数据所有者即为每个车辆。由于车辆本身拥有唯一标示车辆的 DOGID,在上传数据时会将自身的 DOGID 和数据一起上传至云端进行存储和处理。

CSP:首先,对于用户上传数据时附带的 DOGID 进行分析。如果用户在预先规定的可以存储在云设备中的用户列表中,则为用户开辟空间,用以存储上传的数据内容;如果不在列表中,则阻止数据的上传。其次,对于可供外部访问的数据,提取数据内容的 DGID,存储在 DGIDD 中,以便在访问时进行匹配,决定数据是否可以流出云端。

DGIDD:存储云中数据的 DGID,以便在数据使用者 DU 进行数据访问时进行比对和校验,决定是否允许 DU 访问数据。

DU:根据自身的需求产生数据访问请求。

1.3 系统方案描述

整个系统分为四部分:数据所有者 DOGID 的产生与数据的上传,数据存储判断,数据 DGID 的产生与存储,授权数据使用者访问数据。

(1)数据所有者 DOGID 的产生与数据的上传:可由车载终端实现,为车辆网中的每一车辆获取唯一的网络基因标示—“网络车牌”。车载终端采用了模块化设计理念,由智能芯片、中央处理模块、通信模块、定位模块等组成。同时内部嵌有多种传感器,可对几乎所有车辆的静态、动态信息进行感知和监控。这些状态涉及车辆行驶和车体,动力,车辆安全,环境等与具体车辆相关的一些属性。车载终端从汽车和用户中提取天然属性信息,采用“数字基因”技术,形成唯一标示单辆汽车的 ID,使得每辆汽车在网络中都有其唯一的身份标识。这个标识不仅是一个标签,而且是网络可信标识,使得车辆的身份能主动、唯一地被识别。将

车辆在运行中产生的相关数据,如车辆行驶过程中的历史轨迹、访问的路线等,进行动态实时采集,实现车辆驾乘等智能感知。并将数据上传至云端进行存储和处理。

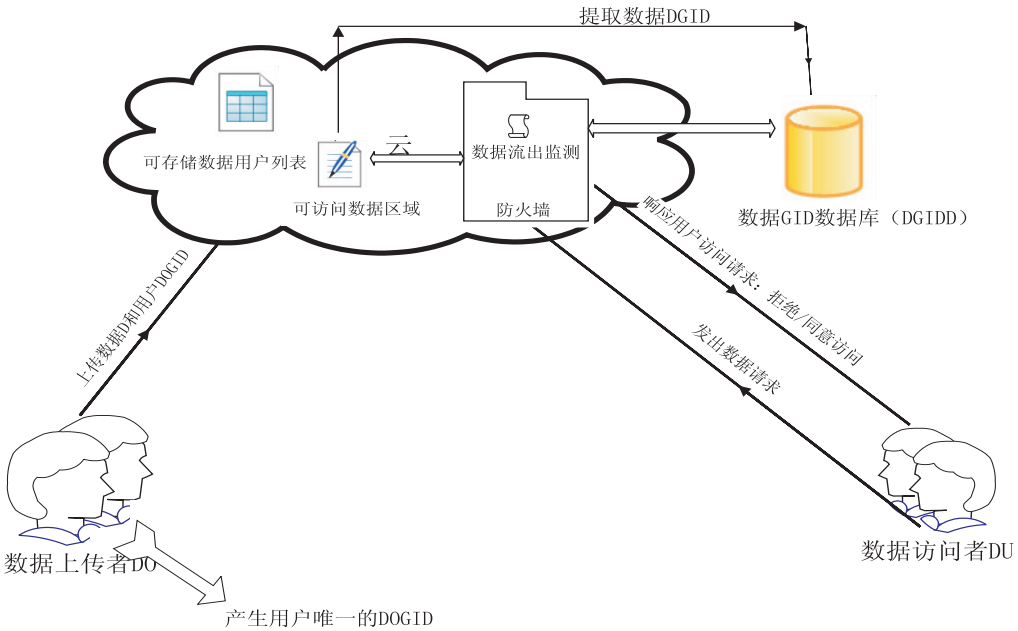

图 1　系统总体架构图

(2)数据存储判断:对于处理特定区域数据的云存储设备,在云端有列表,存储可以将数据放置在此云设备的用户列表。用户 DO′上传数据时,首先对 DO′的 DOGID′在用户列表中查找。如果用户在列表内,则对用户开辟存储空间,进行数据的接收、存储,以便数据的处理和数据使用者对数据进行访问;如果用户不在存储列表中,则将此次数据上传任务取消。

(3)数据 DGID 的产生与存储:在产生 DGID 时,系统首先以文件的重要性作为输入,判断哪些数据可以流出云端,哪些不可以流出云端。对于可以流出云端的数据提取 DGID,存储在 DGIDD 中,当用户访问的数据流的 DGID′与 DGIDD 中的数据流的 DGID 一致时,才允许数据流流出云端。

定义检测时延,数据用户访问时,在被禁止访问时,已经访问的数据比特数和数据用户实际上可以访问的比特数之差,记为 L。很显然,L 越小越好。为达到设定的 L 比特检测时延,数据的 DGID 应该在文件的每 L byte 产生一个 $DGID_i$,$DGID_i$ 的产生采用 128 位的 CRC。随着文件的增大,DGID 的数目不断增大,因为每一个 $DGID_i$ 要包括从文件开头到当前位置的所有 byte 信息。具体的实现步骤如下:

S1:对于每一个给定的数据内容 D,将 D 分为 n 部分,每一部分为 L byte。

S2:128 位的 CRC_i 是第 i 部分的校验码。假设 CRC_1 以 0 为种子数,包括文件的第一部分。CRC_{i+1} 将 CRC_i 作为种子数,计算第 $i+1$ 部分的校验码。以此类推,计算 CRC_i。

因此,前 $i+1$ 部分的数据内容变化将会导致 CRC_{i+1} 部分的校验码发生变化。

(4)授权数据使用者访问数据:数据使用者的访问授权主要是在云端将数据使用者访问请求对应的字符串流与存储在 DGIDD 中的可允许流出的 DGID 进行比较。如果两者相同,则允许数据流出云端,供用户访问;反之,则中断数据链接,禁止数据流出云端。记 t_k 为整数,$k \in [0,p]$ 且 $1 = t_0 \leqslant t_1 \leqslant \cdots \leqslant t_p$,作为监测点。

具体检测方案如下:

(1)对于 D 中的每个字符串 F_j,为 $F_j[t_k]$ 计算 DGID,记作 $DGID[F_j[t_k]]$;

(2)将 D 中子串产生的 DGID 存储在哈希表中;

(3)待访问的数据字符串为 S,计算 $S[t_k]$ 的 DGID,记作 $DGID[S[t_k]]$;

(4)将 $DGID[S[t_k]]$ 存放在哈希表中,如果哈希表有一个或多个入口,则进行比较,查看是否在哈希表中有完全匹配的 DGID;

(5)如果没有 DGID 匹配,将该字符串不属于可以流出云端的数据串,数据传输中止,用户不得访问这个字符串。

2　安全性分析及仿真

2.1　实验环境配置

文件类型选择:使用现实世界的文件集合(总共 70 GB)对该方案的性能进行评估,如表 1 所示。

表 1 实验所用文件集合

文件类型	文件数目	大小/G	日期
txt	454 000	9.3	2008
html	161 000	23.1	2009
混合	20 000	36.8	2011

txt 和 html 文档是由文献[10-11]提供。其中,txt 文件是来自社交网站(del. icio. us)的 454 000 名用户的已经公开的书签中获得,通过从一个小组随机用户开始执行广度优先搜索(BFS)来获得这些用户,进一步获得用户的 txt 书签文件。由于 Wikipedia. org 只保存每个页面的前 500 次修改,因此,通过对 Wikipedia. org 网站页面的前 500 次修改内容获得 html 文件。这些 html 文件仍然通过 BFS 随机选择一个小组常用页面获得。混合文件则通过对“Science”,“Microsoft”,“Technology”等关键字进行谷歌搜索,获得的搜索结果 docx,jpg,pdf 等作为实验测试文件类型。除此之外,为进一步丰富混合文件中的文件类型,将 8 000 个私人 mp3 文件也导入到混合文件中。

实验情形:利用每份文件中的“坏字节”模拟数据泄露。检测时延通过下述方式获得:随机选择 1 000 份文件,在每份文件中选择一个字节作为“坏字节”,对其数值进行修改。于是,对每份文件,对于每个检测位置逐步计算 DGID,检测该 DGID 在哈希表中位置是否存在。如果检测失败,则认为存在坏字节。其中,检测位置和“坏字节”位置间的差值就是检测延时。

2.2 实验结果与分析

数据上传时,由于使用江苏迪纳科技有限公司的 GID 设备产生了自身的 DOGID,这样,上传数据时,云端可以利用这个 DOGID,检测该用户是否是允许存放数据的用户。如果用户在允许存放数据用户列表中,云端会给用户开辟空间,进行数据的存放。这样,可以节约云端内部的存储空间,防止空间的浪费;同时,阻止其他非法用户上传数据,也会进一步加大授权用户数据上传的速率。

在数据访问时,更关注的是在一定的检测时延下内存的使用情况,以及不同内存下的实际检测时延与真实检测时延之间的差值。这是对系统的安全性能的两个主要评价指标。文中将从以上两方面进行仿真分析。

(1)内存使用量。

图 2 给出了不同检测时延期望时,基于 GID 车辆网数据安全的内存使用量。其中,x 坐标和 y 坐标为自然对数坐标(文中的 GID 匹配假设哈希表的利用率为 100%)。

在图中可以看到,随着检测时延的对数增大,内存的对数使用量线性减小。换句话说,不管有多少个监测点,内存使用量和检测时延都是固定的。原因如下:在基于 GID 的匹配算法中,检测点是以固定间隔放置,且间隔长度等于检测时延。对每个检测点,每个文件存储了一个 128 位的 CRC。当检测时延期望增长 Y 倍时,内存数量下降到原来的 $1/Y$。即,对不同的检测时延,每个哈希表的大小是相同的。文中已通过实验证明了这一点,由于篇幅所限,没有给出相关图形。

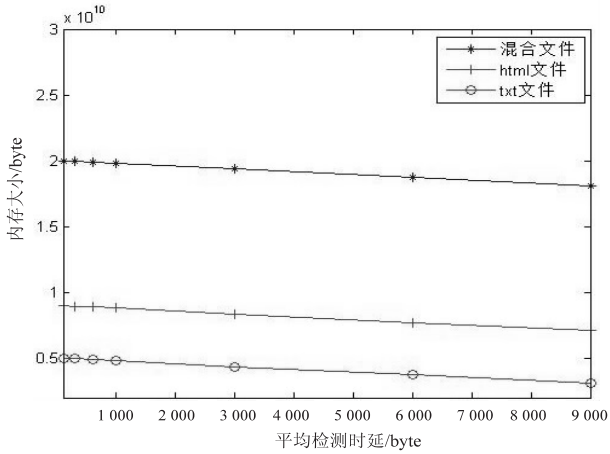

图 2 采用 GID 方案所需的内存使用量

请注意,算法与文献[12]中的 Glavlit 算法相类似,其主要区别在于,Glavlit 算法 使用 160 比特的 SHA-1 提取指纹,而不是文中基于 GID 方案的 128 位 CRC,所以 Glavlit 算法消耗的内存量将多于文中基于 128 位 CRC 的 GID 方案。

(2)实际检测时延。

在 txt,html 和混合三种不同类型的文件集进行仿真。由于结果是相似的,因此只给出 html 文件的预期检测时延和实际检测时延的比较结果,如图 3 所示。

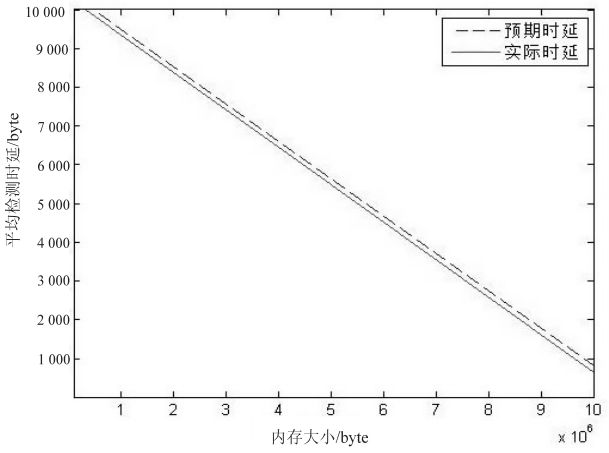

图 3 html 文件预期时延与真实时延之间的比较

可知,实际检测时延总是低于预期检测时延。原因如下:在计算检测延时期望时是假设“坏字节”刚好加在上一个子字符串结束之后,但是在仿真实验中,

(下转第 131 页)

缩短了软件开发时间,提高了软件开发质量,具体数据如表 1 所示。

表 1　应用两种不同模型比较

模型类型	开发人员/人	开发时间/天	代码行数/KLOC	软件大小/GB
传统的三层模型	5	228	15.16	3.25
改进的分层模型	5	137	8.84	1.78

而且,所开发的平台符合当前古建筑爱好者的需求,让爱好者们足不出户就能通过互联网欣赏到自己喜欢的古建筑文物。这不仅对古建筑文物起到了推广宣传作用,而且也为保护和传承古建筑文物奠定了基础。

参考文献:

[1] 宋晓红,袁　慧. 三维激光扫描技术在古建筑文物保护中的应用研究[J]. 测绘技术装备,2014,16(3):40-42.

[2] 闫宏印,张卫争,刘超慧. 开源框架下 Web 应用分层的设计与实现[J]. 计算机工程与设计,2008,29(23):6023-6025.

[3] 谌湘倩,狄文辉,孙　冬. 基于 SSH 框架与 AJAX 技术的 Java Web 应用开发[J]. 计算机工程与设计,2009,30(10):2590-2592.

[4] 李守振,张南平,常国锋. Web 应用分层与开发框架设计研究[J]. 计算机工程,2006,32(22):274-276.

[5] 李　璟. 基于. NET 的分层架构及抽象工厂模式在 Web 开发中的应用[J]. 软件导刊,2015,14(4):105-108.

[6] 周相兵,兰青青,江瑜清. 基于分层结构的 Web 服务与 Ajax 整合的中间件实现研究[J]. 计算机应用与软件,2008,25(11):97-99.

[7] Ren Yongchang, Xing Zhaofeng, Xing Tao, et al. Application research for integrated SSH combination framework to achieve MVC mode[C]//Proc of international conference on computation and information sciences. [s. l.]:[s. n.],2011:499-502.

[8] Ajax Reference Documentation. Introduction to the Ajax[EB/OL]. 2015. http://www. okajax. com/.

[9] Struts Reference Documentation. Introduction to the Struts framework [EB/OL]. 2015. http://struts. apache. org/docs/version-notes-218. html.

[10] 杨　力,陈利学,赵永清,等. 基于移动代理的 Struts2 框架[J]. 计算机工程,2013,39(1):260-263.

[11] Spring Reference Documentation. Introduction to the Spring framework[EB/OL]. 2015. http://spring. io/docs/reference.

[12] Song Hongwei, Liu Xuning, Lu Aiqin, et al. Design and development of practical course experiment management system [C]//Proc of the 8th international conference on computer science & education. [s. l.]:[s. n.],2013:1217-1220.

[13] Hibernate Reference Documentation. Introduction to the Hibernate framework [EB/OL]. 2015. http://hibernate. org/orm/documentation.

[14] 沈　磊. 基于 Struts2 和 Hibernate 的 RBAC 模型设计与实现[D]. 南京:南京师范大学,2011.

(上接第 104 页)

"坏字节"的位置是从均匀分布中随机选择的。

3　结束语

文中利用 GID 设计适合车联网的云端数据安全方案,采用网络基因的方式,保障用户数据在上传、存储和访问时的安全。通过真实数据的仿真结果证明,该方案可以有效减小数据非法访问的概率,同时云存储设备的空间得以充分利用,但是利用哈希表对提取 GID 直接进行匹配比较会导致较大的内存开销。因此,下一步的研究重点是如何减小基于 GID 的车联网云端数据安全方案的内存和计算开销。

参考文献:

[1] 王建强,吴辰文,李晓军. 车联网架构与关键技术研究[J]. 微计算机信息,2011,27(4):156-158.

[2] 唐　伦,柴　蓉,戴翠琴,等. 车联网技术与应用[M]. 北京:科学出版社,2013:96-116.

[3] 王建强,李世威,曾俊伟. 车联网发展模式探析[J]. 计算机技术与发展,2011,21(12):235-238.

[4] 常促宇,向　勇,史美林. 车载自组网的现状与发展[J]. 通信学报,2007,28(11):116-126.

[5] 刘南杰. 崛起中的车联网[J]. 营赢,2011(11):18-23.

[6] Cisco. Cisco ironport data loss prevention[EB/OL]. 2013. http://www. ironport. com/kr/technology/ironport_dlp_overview. html.

[7] Mydlp. Web-based data leakage prevention[EB/OL]. 2014. http://www. mydlp. com/.

[8] Hao F, Kodialam M, Lakshman T V, et al. Protecting cloud data using dynamic inline fingerprint checks[C]//Proc of INFOCOM. Turin:IEEE,2013:2877-2885.

[9] 王志文,王　强. 云计算敏感数据防泄露技术研究[J]. 信息安全与通信保密,2013(8):85-87.

[10] Puttaswamy K, Sala R, Zhao B Y, et al. Starclique: guaranteeing user privacy in social networks against intersection attacks [C]//Proc of CoNEXT. [s. l.]:[s. n.],2009.

[11] Puttaswamy K, Marshall C, Subramanian V R, et al. Docx2go: collaborative editing of fidelity reduced documents on mobile devices[C]//Proc of MobiSys. [s. l.]:[s. n.],2010.

[12] Schear N, Kintana C, Zhang Q, et al. Glavlit: preventing exfiltration at wire speed[C]//Proc of HotNets. [s. l.]:[s. n.], 2006.