doi:10.3772/j.issn.1002-0470.2020.04.006

基于属性加密的软件定义网络域间访问控制方法①

周 波②* 王树磊**

(*南京信息职业技术学院电子信息学院 南京 210023)

(**常州工学院民航飞行学院 常州 213032)

摘 要 软件定义网络(SDN)区别于传统的计算机网络,其核心思想是将网络控制与数据转发分离。SDN 不仅有利于降低网络当中的硬件成本,还使得网络管理员能够方便地对来自不同厂商的设备进行集中化的调试和管理。尽管具备传统网络不可比拟的优势,SDN 的应用却带来了新的安全问题。当前如何保证 SDN 控制器掌控的敏感信息不被窃取是 SDN 安全领域内的关键问题之一。然而现有 SDN 访问控制方案往往无法提供安全的信息保护,而且往往着眼于单域环境下的管理,不能满足多域网络管理的需求。本文提出了一种抗密钥暴露的密文策略属性快速加密算法(AKE-CP-ABFE),基于该算法构建了一种针对多控制器设置的 SDN 域间访问控制系统模型。利用预计算技术使加密者无需进行复杂的指数运算或双线性映射。此外将用户或者设备的 MAC 地址嵌入到私钥当中,即使私钥已经暴露,获取私钥的人也无法使用该私钥。分析证明 AKE-CP-ABFE 具备良好的安全性。仿真实验表明,该方法能以较高的计算效率保证 SDN 敏感信息域间分享的安全性和灵活性。

关键词 软件定义网络(SDN);域间访问控制;属性加密(ABE);预计算;抗密钥暴露

0 引 言

软件定义网络[1](software defined network,SDN)是一种区别于传统计算机网络的新型网络架构模式,自提出以来得到了学术界乃至工业界的广泛关注。SDN 吸取了 ForCES 架构[2]的思想,将网络的控制层与数据层分离,该设计不仅有利于降低网络当中的硬件成本,还使得网络管理员能够方便地对来自不同厂商的设备进行集中化的调试和管理。相比传统计算机网络,SDN 具备前者所没有的性能优势,但 SDN 的一系列安全问题[3]却成为了阻碍其进一步广泛应用的难题。其中最关键的问题之一就是,如何在远程控制环境下保证由 SDN 控制器掌控的用户和设备敏感信息不被攻击者窃取[4]。SDN 的控制器负责管理流表,而流表包含敏感的控制信息,对于数据的转发起到了决定性的作用。然而当前的 SDN 架构允许应用程序接口轻易地调用流表而无需权限认证。更为严重的是相当一部分接口具备公开信息的权限。因此 SDN 控制器必须保证只有经过授权的设备才能获取以上的敏感信息。如今许多大型的 SDN 架构都采用了多控制器管理域间通信,控制器除了与数据转发设备进行交互,还要与各个控制器之间保证规则统一与信息同步。因此在 SDN 当中必须采用一种机制,该机制不仅能够保证 SDN 流表的安全性还必须对流表的访问权限进行灵活、细粒度的控制。

Ethane 系统[5]作为 SDN 前身提供了集中化的

① 国家自然科学基金(61571241)和江苏高校品牌专业建设工程项目(PPZY2015C242)资助。

② 男,1978 年生,硕士,讲师;研究方向:信息安全,无线传感网;联系人,E-mail:bozhou_njcit@126.com

(收稿日期:2019-04-18)

网络安全管理方案。该系统利用规定的安全信道实现控制信息在控制器和交换机之间的传输,但其过于简单,无法满足在大规模部署时的安全需求。文献[6]根据 OpenFlow 规则说明书提出,控制器与数据转发设备间必须通过传输层安全协议(transport layer security protocol,TLS)实现双向认证,以避免非法用户伪装成控制器与交换机进行通信。然而并没有规定域间访问的控制方法,这使得大型 SDN 架构有可能面临未知的非法规则插入或规则修改操作。文献[7]提出了一种针对 SDN 控制器的访问控制系统。为实现不同 SDN 组件访问权限的配置,该系统首先将 SDN 的各个组件在逻辑上进行隔离,其次对不同组件的访问权限进行详细的描述,然后在用户试图对组件进行访问和控制时必须对用户进行身份认证。此外,该方案配置了一种用户访问冲突的解决机制,保证多个用户可以同时访问同一个组件。文献[8]提出了一种面向 SDN 控制层和应用层的访问控制方案,该方案对不同网络组件的可执行指令做出了不同程度的限制,使得在云计算环境下实现安全、快速地访问控制。Kamath 等人[9]设计了一种 SDN 认证架构,该架构提供一种灵活的身份认证和访问控制方法,同时将所有未授权的设备隔离起来以保证网络资源的安全性。尽管在访问控制领域有大量的研究成果,但是 SDN 控制器始终面临的是一种动态变化的网络结构,这使得 SDN 控制器在执行访问控制操作时必须具备相当的灵活性、安全性以及高效率。这对现有的访问控制理论提出了严峻的挑战,然而现有的访问控制方法几乎无法满足这样的需求。

属性加密[10](attribute-based encryption,ABE)是近些年来提出的一种功能加密算法,它能够从密码学原理上填补以上的空白。ABE 为数据加解密操作赋予了一套访问策略以及具备一定描述性的属性集合。只要属性集合与访问策略的相似度超过事先设定的阈值,那么解密者就可以成功解密。具体来说目前有 2 种主要的 ABE 算法。一种是基于密钥策略的 ABE 算法[11](key policy ABE,KP-ABE),它将访问策略嵌入到密钥当中,而属性集合则嵌入到密文当中,最早提出的 ABE 算法正是这种 KP-ABE;一种是基于密文策略的 ABE 算法[12](ciphertext policy ABE,CP-ABE),它将访问策略嵌入到密文当中,而属性集合则嵌入到密钥当中。由于 CP-ABE 算法允许加密者自由制定访问策略,而密钥当中嵌入的属性集合又能够表征不同解密者的身份,因而 CP-ABE 更适合构建一种安全又灵活的云存储访问控制算法。文献[13]提出了一种基于多属性权威的 ABE 算法,每个属性权威拥有各自的主密钥,这使得单个属性权威不再担负过重的计算任务。文献[14]提出了一种基于双向安全计算产生主密钥的机制,同时设计了一种高效的撤销方法,在提高计算效率的同时也丰富了 ABE 的功能。文献[15]采用零知识证明使 2 个属性权威互动产生主密钥,从而有效防止密钥托管问题的产生,同时支持密钥追责防止用户将自己的私钥交由他人使用。文献[16]提出了一种 CP-ABE 的属性直接撤销方法保证了用户更新的灵活性,同时使得其密文、私钥和公钥的长度都有所优化。文献[17]提出了一种不需要进行配对运算的 ABE 算法,但是由于使用阈值门作为访问策略,因此算法整体显得不够灵活。文献[18]提出了一种分布式的 CP-ABE 密钥管理协议,同时利用外包解密使得终端用户无需进行任何配对计算就可以进行解密,但是算法整体的计算负载仍然较大。

ABE 算法在大型访问控制系统上的应用潜力已被相关领域学者所认可,但是其复杂的计算将造成系统运算负载急剧上升,现有的 ABE 算法还远不能达到在 SDN 中实际部署的水平。尤其在大型 SDN 环境下,来自各类厂商的设备性能不同,如果性能稍有限制,就不能保证 SDN 访问控制的灵活性、即时性和安全性。文献[19]提出了在 SDN 环境中应用属性加密算法并提出了一种基于等级 CP-ABE 的 SDN 访问控制方案,使得 SDN 控制器能够灵活管理 SDN 的用户、设备和流表数据。文献[20]提出基于代理群的 ABE 算法,把复杂计算全部交给可信的代理节点群处理。虽然降低了 ABE 算法的计算门槛,但是系统整体的安全性不再单纯依赖于 ABE 算法的安全性,而在某种程度上几乎依赖于代理节点的可信等级。此外,代理节点群的存在使得

加解密过程中产生了额外的通信开销。

本文结合 SDN 实际环境在现有的 CP-ABE 方案基础上对算法的安全性、计算效率进行了改进。提出了一种抗密钥暴露的密文策略属性快速加密算法(anti-key-exposure ciphertext policy attribute-based fast encryption, AKE-CP-ABFE),首先利用预计算技术(pre-computation technique, PCT)将必要的元素计算出来并存储起来,加密者在进行加密时无需进行任何复杂的指数运算或双线性映射,提高了属性加密的计算效率。同时基于拓展图技术(expander graph technique, EGT)减少了预计算产生的元素数量,进一步降低了预计算产生的存储开销。除此之外,本文将用户或者设备的 MAC 地址嵌入到私钥当中,如果私钥当中的地址信息与设备本身不符,则无法完整解密。这保证了即使将私钥有意或者无意地暴露给其他非法用户或者设备,他们也无法获取此私钥获取敏感信息。理论分析证明 AKE-CP-ABFE 具备良好的安全性。基于该算法构建了一种针对多控制器设置的 SDN 域间信息访问控制系统模型,该模型降低了开源 API 造成的用户或者网络设备敏感信息泄露的风险。仿真实验表明,该系统能够在部署多控制器的大型 SDN 环境下以较高的计算效率保证敏感信息域间分享的安全性和灵活性。

1　预备知识

1.1　单调访问策略

设 $\{P_1, P_2, \cdots, P_n\}$ 是由若干元素组成的集合,访问策略$\mathbb{A}$是集合 $\{P_1, P_2, \cdots, P_n\}$ 的非空子集,即$\mathbb{A} \in 2^{\{P_1, P_2, \cdots, P_n\}} \backslash \varnothing$。若对于任意 2 个集合 B 和 C, 当满足 $B \in A$ 并且 $B \subseteq C$ 时,使得 $C \in \mathbb{A}$,那么称$\mathbb{A}$是单调访问策略。

若满足 $S \in \mathbb{A}$,则集合 S 是关于访问策略$\mathbb{A}$的合法集合,反之,则称 S 为非法集合。

1.2　访问树

令 Γ 是一个树形的访问策略。Γ 中每一个节点由一个阈值门表示,阈值门由该节点的子节点个数及一个阈值组成。设任意节点 x 的子节点个数为 num_x, 阈值为 k_x, 那么 $0 \leqslant k_x \leqslant num_x$。当 $k_x = 1$ 时该阈值门表示的是一个“或”门,而当 $k_x = num_x$ 时表示的是一个“与”门。对于 Γ 当中的叶节点,其表示的是一个属性,此类节点的阈值是 1。为便于展开讨论,定义以下函数:

(1) $parent(x)$: 返回节点 x 的父节点;

(2) $att(x)$: 返回叶节点 x 所代表的属性;

(3) $index(x)$: 返回节点 x 的索引号,访问树 Γ 为每一个节点设置了唯一的索引号。

对于一个属性集合 S 和访问树 Γ,若 S 满足 Γ 则表示为 $\Gamma(S) = 1$。通过如下的迭代方法来判断 S 是否满足 Γ:假设 R 是访问树 Γ 的根节点, Γ_x 是关于节点 x 的子树。如果 x 是非叶节点,设其任意的子节点为 z 并首先计算 $\Gamma_z(S)$。当且仅当 k_x 个子节点的返回值为 1 才使得 $\Gamma_x(S) = 1$ 成立。若 x 是叶节点,当且仅当 $att(x) \in S$ 才使得 $\Gamma_x(S) = 1$ 成立。

1.3　双线性映射

设$\mathbb{G}_1$和$\mathbb{G}_2$是 2 个阶为大素数 p 的循环群, g 是 $\mathbb{G}_1$ 的一个生成元,映射 $e: \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ 是关于 $\mathbb{G}_1$和$\mathbb{G}_2$的双线性映射,当且仅当 e 满足以下性质:

(1) 双线性:对于任意的 $u, v \in \mathbb{G}_1$以及 $a, b \in Z_p$, 都有 $e(u^a, v^b) = e(u, v)^{ab}$;

(2) 非退化性:存在生成元 g, 使得 $e(g, g) \neq 1$, 其中 1 是$\mathbb{G}_2$的单位元;

(3) 可计算性:对于任意的 $u, v \in \mathbb{G}_1$,存在多项式时间算法能够有效计算出 $e(u, v)$ 的值。

1.4　预计算

由于即时的双线性映射会消耗相当多的计算资源,这直接成为了 ABE 算法在效率上难以提高的瓶颈。虽然也有学者提出了去双线性化的思想,但是去双线性化后的安全性始终无法与原有的 ABE 算法相提并论。双线性映射预计算的基本思想是将一系列双线性映射预先计算并存储起来,因此在 ABE 算法上采用预计算方法能够在运算负载、存储负载以及算法安全性上达到更好的平衡,从而使得 ABE 算法更适用于构建 SDN 域间信息访问控制系统。

预计算的基本思想是,首先产生一个椭圆曲线上的生成元 P, 其次提前计算好 n 个元组 (r_i, r_iP)。在此基础上,每一个新的元组都由 k 个已有的元组共同产生,即:

$$r = \sum_{1 \leqslant i \leqslant k} r_i \bmod p$$

$$rP = \sum_{1 \leqslant i \leqslant k} r_i P \bmod p$$

也就是说，通过 $k-1$ 次模加操作来代替相对复杂的椭圆曲线点乘操作。为了进一步提高计算效率，本文采用了一种基于拓展图的预计算方法来降低参数 k 的值。

1.5 判定双线性 Diffie-Hellman 困难假设

本方法的安全性建立在判定双线性 Diffie-Hellman 困难假设（decisional bilinear Diffie-Hellman assumption，DBDH）上。

设$\mathbb{G}_1$是一个根据安全参数生成的阶为素数 p 的群。产生 3 个秘密的随机数 a，b，$c \in Z_p$。若敌手已知一组参数 $\{P, aP, bP, cP\}$，那么该敌手难以区分 $e(P, P)^{abc}$ 与 $e(g, g)^z$。若存在一个敌手 A 在获得参数 $\{P, aP, bP, cP, Y\}$ 后，输出 1 表示 $Y = e(P, P)^{abc}$，反之输出 0 表示 $Y = e(P, P)^z$。敌手 A 能够以优势 ε 解决 DBDH 问题，当且仅当：

$$\left| \begin{aligned} &\Pr[A(P, aP, bP, cP, e(P, P)^{abc}) = 1] \\ &- \Pr[A(P, aP, bP, cP, e(P, P)^z) = 1] \end{aligned} \right| \geqslant \varepsilon$$

2 AKE-CP-ABFE 模型

2.1 模型内交互角色

AKE-CP-ABFE 模型涉及以下 4 类交互角色。

（1）属性权威：是一个可信的权威机构，负责所有属性的认证以及公钥私钥的发布。

（2）SDN 控制器：负责收集、存储和管理 SDN 流表、路由以及数据量等重要信息，其中包含各类用户或者设备的敏感信息。本方案针对大型 SDN 采用多控制设置，将 SDN 划分为多个 SDN 域，每一个域内部署唯一的 SDN 控制器。每个 SDN 控制器管理各自域内的重要信息，同时负责与其他域的 SDN 控制器交互。

（3）加密者：是数据的初始拥有者（可以是用户，也可以是产生数据的 SDN 设备），能够上传自己的数据并根据自己的意愿或需求自由制定相应的访问策略。

（4）解密者：是试图获取 SDN 控制器内信息的用户或者设备，其身份用一个属性集合来表示。解密者拥有一个与其属性集合相对应的私钥，并通过私钥来解密密文。

2.2 算法框架

AKE-CP-ABFE 算法包括初始化算法、私钥生成算法、预计算算法、快速加密算法和解密算法共 5 个主要算法，算法框架如下：

（1）初始化算法：由属性权威执行。算法输入一个安全参数 k，输出公钥 PK 和主密钥 MSK，其中公钥 PK 向全网公开，而主密钥 MSK 由属性权威秘密保存。

（2）私钥生成算法：由属性权威执行。解密者发起密钥生成请求时输入其唯一的 MAC 地址 A_{MAC}、属性集合 S、公钥 PK 以及主密钥 MSK，最终输出与其 MAC 地址以及属性集合相对应的私钥 SK。

（3）预计算算法：由属性权威执行。算法由 2 个子算法组成，分别是预处理子算法以及元组生成算法。预处理子算法输入公钥 PK，输出两组列表 L 和 L'。元组生成算法输入一个访问树 Γ，输出与访问树 Γ 相对应的元组 $Tuple_1$ 和 $Tuple_2$。

（4）快速加密算法：由加密者执行，算法首先输入一个访问树 Γ、公钥 PK 以及消息明文 M，然后通过调用预处理算法输出与访问树 Γ 对应的密文 CT，最后将密文 CT 上传至域内的 SDN 控制器。

（5）解密算法：由解密者执行，输入其 MAC 地址 A_{MAC}、密文 CT 以及私钥 SK，当解密者的属性集合满足访问策略，同时其 MAC 地址与隐藏在私钥 SK 当中的 MAC 地址信息相吻合时，才会输出消息明文 M，否则退出执行。

基于以上算法本文构建了面向多控制器 SDN 的域间访问控制系统模型。该模型在 2 个 SDN 域之间进行信息分享的工作流程如图 1 所示，系统启动时属性权威执行初始化算法生成公钥 PK 和主密钥 MSK。与此同时，属性权威执行预计算算法生成一系列参数并存储在列表 L 和 L' 当中用于后续的快速加密。当由解密者请求生成私钥时，属性权威执行密钥生成算法产生与其属性集合 S 以及其 MAC 地址 A_{MAC} 相关的私钥 SK。对于加密者产生的流表等各类信息，加密者制定相应的访问树 Γ 并通过快速加密算法生成密文 CT 并将密文 CT 上传给

当前 *SDN* 域的控制器。对于不同域的解密者,当其发出解密请求时,当前 SDN 域控制器首先将密文 *CT* 发送给另一个 SDN 域控制器,然后将密文 *CT* 转发给解密者。如果此时解密者的属性集合 S 满足密文当中的访问树 Γ,而且解密者本身的 MAC 地址 A_{MAC} 与私钥 *SK* 当中嵌入的 MAC 地址信息吻合,那么解密者才能获取该信息的访问权限。反之则无法获取任何有用的信息。

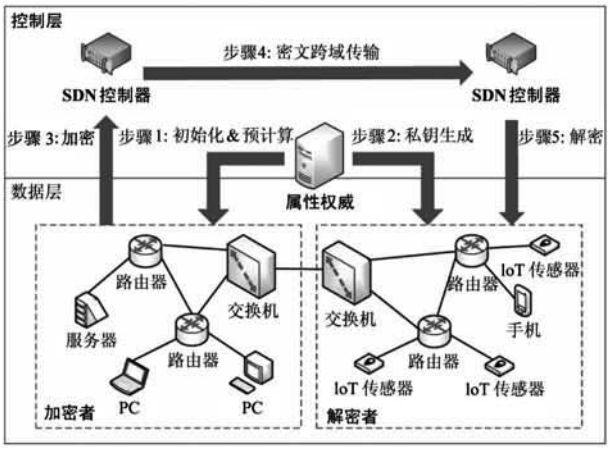

图 1　基于 AKE-CP-ABFE 的 SDN 域间访问控制系统模型

3　AKE-CP-ABFE 算法流程

为方便算法构建,首先假设$\mathbb{G}_1$和$\mathbb{G}_2$是阶为大素数 p 的双线性群,P 是$\mathbb{G}_1$的一个生成元,$e:\mathbb{G}_1\times\mathbb{G}_1\to\mathbb{G}_2$是群$\mathbb{G}_1$到群$\mathbb{G}_2$的双线性映射。取一个安全参数 k,这个参数决定了大素数 p 的比特位数。然后定义拉格朗日函数$\Delta_{i,S}(x)$,使得集合 S 当中的任意一个元素 i 都有:

$$\Delta_{i,S}(x)=\prod_{j\in S,j\neq i}\frac{x-j}{i-j}$$

此外定义 2 个哈希函数 $H_1:\{0,1\}^*\to\mathbb{G}_1$和 $H_2:\{0,1\}^{48}\to Z_p$。其中 H_1 将任意一串二进制数组映射为群$\mathbb{G}_1$中的元素,H_1 将 48 bit 的 MAC 地址映射为 Z_p 上的元素。

3.1　初始化算法

初始化算法首先输入全局的属性集合 $\Omega=\{att_1,att_2,att_3,\cdots,att_n\}$ 以及一个安全参数 k,根据参数 k 生成 2 个阶为 p 的双线性群$\mathbb{G}_1$和$\mathbb{G}_2$,其次选择$\mathbb{G}_1$的一个生成元 P 以及一个双线性映射 $e:\mathbb{G}_1\times\mathbb{G}_1\to\mathbb{G}_2$。然后定义 2 个哈希函数 $H_1:\{0,1\}^*\to\mathbb{G}_1$和 $H_2:\{0,1\}^{48}\to Z_p$,并选择 2 个随机数 $\alpha,\beta\in Z_p$。最后生成如下公钥:

$$PK=\{p,P,e,\Omega,e(P,P)^{\alpha},H=\beta P,T=(\frac{1}{\beta})P,H_1,H_2\}$$

同时保存如下的主密钥:

$$MSK=\{\alpha P,\beta\}$$

3.2　私钥生成算法

私钥生成算法首先输入公钥 *PK*、主密钥 *MSK*、一个属性集合 S 以及一个 MAC 地址 A_{MAC}。其次选择一个随机数 $r\in Z_p$,然后计算 $D=\frac{\alpha+rH_2(A_{MAC})}{\beta}P$。对于任意一个属性 $att_j\in S$ 选择一个随机数 $r_j\in Z_p$ 并计算 $D_j=(rP)\cdot(r_jH_1(att_j))$ 和 $D'_j=r_jP$。最后输出关于属性集合 S 的私钥:

$$SK=\{S,D=((\alpha+rH_2(A_{MAC}))/\beta)P,\forall att_j\in S:D_j=(rP)\cdot(r_jH_1(att_j)),D'_j=r_jP\}$$

3.3　预计算算法

预计算算法由 2 个子算法组成,分别为预处理算法和元组生成算法。为方便算法描述,本文定义了一些变量,变量定义说明如表 1 所示。

表 1　变量定义说明

变量	定义说明
Y	访问树 Γ 的叶节点集合
U_x	子树 Γ_x 包含的属性的集合
S_x	节点 x 的子节点集合
S'_x	节点 x 的子节点索引集合
$c(\varepsilon)$	ε 拓展图常数

预处理算法　预处理算法的执行流程如下。

(1) 生成 n 个随机数 $r_1,r_2,\cdots,r_n\in Z_p$;

(2) 对于 $\forall i\in[1,n]$,计算 $e(P,P)^{\alpha r_i}$、r_iP 以及 $\{\forall att_j\in\Omega:r_iH_1(att_j)\}$;

(3) 生成一张空列表 L,对于 $\forall i\in[1,n]$ 将 $\{e(P,P)^{\alpha r_i},r_iP,\{\forall att_j\in\Omega:r_iH_1(att_j)\}\}$ 添加到列表 L 当中;

(4) 计算 $n_e=c(\varepsilon)\log_2(p)$ 并生成 n_e 个随机数 $d_1,\cdots,d_{n_e}\in Z_p$;

(5) 对于 $\forall i \in [1,n_e]$，计算 $e(P,P)^{\alpha d_i}$、d_iP 以及 $\{\forall att_j \in \Omega: d_iH_1(att_j)\}$；

(6) 生成一张空列表 L'，对于 $\forall i \in [1,n_e]$ 将 $\{e(P,P)^{\alpha d_i}, d_iP, \{\forall att_j \in \Omega: d_iH_1(att_j)\}\}$ 添加到列表 L' 当中；

(7) 随机选择 d_i，初始化参数 $t = d_i$、$T_1 = e(P,P)^{\alpha d_i}$、$T_2 = d_iP$ 以及 $\{\forall att_j \in \Omega: T_{3,j} = d_iH_1(att_j)\}$。

元组生成算法 元组生成算法输入一个访问树 Γ，输出与访问树 Γ 相关的 2 个元组。该算法由 A_1 和 A_2 2 个算法组成，算法执行流程如下。

对于访问树 Γ 的根节点，执行 A_1 算法。

(1) 在列表 L' 当中随机选择一组元素 $\{e(P,P)^{\alpha d_i}, d_iP, \{\forall att_j \in \Omega: d_iH_1(att_j)\}\}$；

(2) 重新赋值 $T_1 = T_1e(P,P)^{\alpha d_i}$、$T_2 = T_2 + d_iP$ 以及 $\forall att_j \in U_R: T_{3,j} = T_{3,j} + d_iH_1(att_j)$；

(3) 随机提取一个集合 $S \subset [1,n]$ 使得 $|S| = k$；

(4) 从列表 L 当中随机地抽取一组元素 $\{e(P,P)^{\alpha r_i}, r_iP, \{\forall att_j \in \Omega: r_iH_1(att_j)\}\}$；

(5) 声明并初始化变量 $e(P,P)^{\alpha s} = T_1\prod_{i\in S}e(P,P)^{\alpha r_i}$，如果 $e(P,P)^{\alpha s}$ 是群 $\mathbb{G}_2$ 的单位元则立即返回步骤(1)重新计算；

(6) 声明并初始化变量 $sP = T_2 + \sum_{i\in S} r_iP$ 以及 $\forall att_j \in U_R: sH_1(att_j) = T_{3,j} + \sum_{i\in S} r_iH_1(att_j)$，最终返回元组：

$Tuple_1 = (e(P,P)^{\alpha s}, sP, \{\forall att_j \in U_R: sH_1(att_j)\})$。

对于访问树中的任意节点 x，执行 A_2 算法。

(1) 声明并初始化常量 $d = k_x - 2$；

(2) 从列表 L' 中随机选择一组元素 $\{e(P,P)^{\alpha d_i}, d_iP, \{\forall att_j \in \Omega: d_iH_1(att_j)\}\}$；

(3) 重新赋值变量 $T_2 = T_2 + d_iP$ 以及 $\forall att_j \in U_x: T_{3,j} = T_{3,j} + d_iH_1(att_j)$；

(4) 随机提取一个集合 $S_d \subset [1,n]$ 使得 $|S_d| = k$；

(5) 从列表 L 中随机选择一组元素 $\{e(P,P)^{\alpha r_i}, r_iP, \{\forall att_j \in \Omega: r_iH_1(att_j)\}\}$；

(6) 声明并初始化变量 $c_dP = T_2 + \sum_{i\in S_d} r_iP$，若 c_dP 是群 $\mathbb{G}_1$ 中的单位元则立即返回步骤(1)重新计算；

(7) 从列表 L' 当中随机选择一组元素 $\{e(P,P)^{\alpha d_i}, d_iP, \{\forall att_j \in \Omega: d_iH_1(att_j)\}\}$；

(8) 对于 $\forall u \in [0,d-1]$ 重新赋值变量 $T_2 = T_2 + d_iP$；

(9) 对于 $\forall u \in [0,d]$ 重新赋值变量 $\forall att_j \in U_x: T_{3,j} = T_{3,j} + d_iH_1(att_j)$；

(10) 随机选择一个集合 $S_u \subset [1,n]$ 使得 $|S_u| = k$；

(11) 在列表 L 当中随机选择一组元素 $\{e(P,P)^{\alpha r_i}, r_iP, \{\forall att_j \in \Omega: r_iH_1(att_j)\}\}$；

(12) 对于 $\forall u \in [0,d-1]$ 声明并赋值变量 $c_uP = T_2 + \sum_{i\in S_u} r_iP$；

(13) 对于 $\forall u \in [0,d]$ 声明并赋值变量 $\forall att_j \in U_x: c_uH(att_j) = T_{3,j} + \sum_{i\in S_u} r_iH_1(att_j)$，返回元组：

$$Tuple_2 = \{\forall i \in [0,d]: D_{x,i} = c_iP, \{\forall att_j \in U_x: D'_{x,i,j} = c_iH_1(att_j)\}\}$$

3.4 快速加密算法

为了生成明文 M 与访问树 Γ 相对应的密文，快速加密算法需要对访问树 Γ 当中任意一个节点 x 产生一个次数为 $k_x - 1$ 的随机多项式。为了减小加密的计算负担，利用预计算算法来生成每个节点的随机多项式。根据访问树 Γ 的结构，这种方法采用一种自上而下的迭代方式。

对于访问树的根节点 R，快速加密算法调用 A_1 算法获取元组：

$$(e(P,P)^{\alpha s}, sP, \{\forall att_j \in U_R: sH_1(att_j)\})$$

如果 $k_R - 1 \neq 0$，则继续调用 A_2 算法获取元组：

$$\{\forall i \in [0,k_R-2]: D_{R,i} = c_iP, \{\forall att_j \in U_R: D'_{R,i,j} = c_iH_1(att_j)\}\}$$

利用获取的以上 2 个元组，同时声明并初始化常量 $d_R = k_R - 2$，然后定义一个次数为 d_R 的多项式：

$$r(X) = \sum_{0\leqslant i\leqslant d_R} c_iX^i$$

随后，关于根节点 R 的完整多项式为

$$q_R(X) = r(X)\cdot X + s$$

对于 $\forall l \in S_R$, $\forall j \in U_R$ 计算:

$$q_R(index(l))P = \sum_{0\leqslant i\leqslant d_R} index(l)^{i+1} D_{R,i} + sP$$

$$q_R(index(l))H_1(att_j) = \sum_{0\leqslant i\leqslant d_R} index(l)^{i+1} D'_{R,i,j} + sH_1(att_j)$$

对于访问树 Γ 当中的任意节点 x, 调用 A_2 算法获取元组:

$$\{\forall i \in [0,k_x-2]: D_{x,i} = c_i P,$$
$$\{\forall att_j \in U_x: D'_{x,i,j} = c_i H_1(att_j)\}\}$$

声明并初始化常量 $d_x = k_x - 2$, 利用上式中的系数 c_i 定义一个次数为 d_x 的多项式:

$$r(X) = \sum_{0\leqslant i\leqslant d_x} c_i X^i$$

然后生成关于节点 x 的完整多项式:

$$q_x(X) = r(X)\cdot X + q_{parent(x)}(index(x))$$

对于 $\forall l \in S_x$, $\forall j \in U_x$ 计算:

$$q_x(index(l))P = \sum_{0\leqslant i\leqslant d_x} index(l)^{i+1} D_{x,i} + q_{parent(x)}(index(x))P$$

$$q_x(index(l))H_1(att_j) = \sum_{0\leqslant i\leqslant d_x} index(l)^{i+1} D'_{x,i,j} + q_{parent(x)}(index(x))H_1(att_j)$$

因此对于任意的叶节点 x, 快速加密算法通过迭代得到了 $q_x(0)P$ 和 $q_x(0)H_1(att(x))$。最终生成如下的密文。

$$CT = (\Gamma, \tilde{C} = Me(P,P)^{\alpha s}, C = sH, \forall y \in Y:$$
$$C_y = q_y(0)P, C'_y = q_y(0)H_1(att(y)))$$

3.5　解密算法

首先定义一个迭代函数 *DecryptNode*, 该函数的输入为密文 $CT = (\Gamma, \tilde{C}, C, \forall y \in Y: C_y, C'_y)$、私钥 $SK = \{S, D, \forall att_j \in S: D_j, D'_j\}$ 以及访问树 Γ 当中的一个节点 x。该算法的迭代过程如下:

如果 x 是叶节点,则令 $att_i = att(x)$ 并进行如下的判断与计算。

(1) 若 $att_i \in S$ 则计算并输出以下结果

$$DecryptNode(CT,SK,x) = \frac{e(D_i,C_x)}{e(D'_i,C'_x)} = \frac{e((rP)\cdot(r_i H_1(att_i)), q_x(0)P)}{e(r_i P, q_x(0)H_1(att(x)))} = e(P,P)^{rq_x(0)}$$

(2) 若 $att_i \notin S$ 则输出以下结果表示放弃该节点的计算

$$DecryptNode(CT,SK,x) = \perp$$

如果 x 是非叶节点,那么对于其任意子节点 z 调用函数 *DecryptNode* 并记其返回结果为 F_z。然后进行如下的判断与计算。

(1) 判断是否存在关于节点 x 的包含 k_x 子节点的集合 S_x, 使得 $\forall z \in S_x: F_z \neq \perp$ 成立。若不存在,那么输出 $\perp$ 表示放弃该节点的计算。

(2) 若存在集合 S_x, 计算并输出以下结果:

$$DecryptNode(CT,SK,x) = \prod_{z\in S_x} F_z^{\Delta_{i,S'_x}(0)} = \prod_{z\in S_x} (e(P,P)^{rq_z(0)})^{\Delta_{i,S'_x}(0)} = \prod_{z\in S_x} (e(g,g)^{rq_{parent(z)}(index(z))})^{\Delta_{i,S'_x}(0)} = e(g,g)^{rq_x(0)}$$

到目前为止,本文定义了函数 *DecyrptNode*, 现在进一步定义解密算法。算法首先从访问树 Γ 的叶节点开始调用函数 *DecyrptNode*。如果属性集合 S 满足访问树,那么就可以通过层层迭代获取隐藏在根节点 R 当中的秘密,记作:

$$A = DecryptNode(CT,SK,R) = e(P,P)^{rq_R(0)} = e(P,P)^{rs}$$

随后通过如下计算就可以得到明文:

$$\frac{\tilde{C}\cdot A^{H_2(A_{MAC})}}{e(C,D)} = \frac{Me(P,P)^{\alpha s}\cdot e(P,P)^{rH_2(A_{MAC})s}}{e(sH, \frac{((\alpha + rH_2(A_{MAC}))}{\beta})P)} = M$$

反之,如果属性集合 S 不能满足访问树 Γ,那么将无法在多项式时间内恢复出根节点 R 当中的秘密。与此同时,如果当前的 MAC 地址信息与私钥 SK 当中的 MAC 地址信息不吻合,即使恢复出根节点 R 当中的秘密也无法通过进一步的计算得到消息明文。以上两步中只要任何一步不满足条件,都将导致解密算法退出执行。

4　安全证明与性能分析

4.1　安全证明

本小节给出 AKE-CP-ABFE 的安全证明。通过规约至 DBDH 困难假设,证明了 AKE-CP-ABFE 具备密文的不可区分性。首先给出如下定理。

定理(不可区分性) 给定 2 段长度相同的明

文,如果 DBDH 问题是难解的,那么一定不存在多项式时间内的敌手能够以不可忽略的优势辨别出 AKE-CP-ABFE 的密文来自于哪段明文。

证明 为证明以上定理本文设计了一个挑战游戏,该游戏涉及敌手 A、模拟器 B 以及挑战者 C 3 个角色。

首先,游戏进入初始化阶段,该阶段的流程如下:

(1) 敌手 A 向模拟器 B 发送一个挑战访问树 Γ^*;

(2) 挑战者 C 产生 3 个秘密随机数 $a,b,c \in Z_p$ 和一个双线性群 $\mathbb{G}_1$,然后选择该群上的一个生成元 P,最后将 P、$P_1 = aP$、$P_2 = bP$、$P_3 = cP$ 和 Z 发送给模拟器 B;

(3) 模拟器 B 产生一个随机数 $\beta \in Z_p$、一个双线性群 $\mathbb{G}_2$、一个双线性映射 $e: \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$、一个全局属性集合 $\Omega = \{att_1, att_2, att_3, \cdots, att_n\}$ 以及一个随机预言机 和 $H: \{0,1\}^* \rightarrow \mathbb{G}_1$。最后,模拟器 B 把如下的公钥发送给敌手 A。

$$PK = \{p, P, H = \beta P, T = (\frac{1}{\beta})P, e(P_1, P_2), \Omega, H_1\}$$

其次,游戏进入询问阶段,在该阶段敌手 A 向模拟器 B 进行有限次数的参数询问,参数询问包括私钥询问和加密询问。

私钥询问的流程如下:

(1) 敌手 A 向模拟器 B 发送一个属性集合 S;

(2) 模拟器 B 产生 2 个随机数 $R_1, R_2 \in Z_p$,然后对于属性集合 S 当中的任意属性 att_j 产生一个对应的秘密随机数 $r_j \in Z_p$,最终返回私钥:

$$SK = \{S, D = R_1 P, \forall att_j \in S: D_j = (R_2 P) \cdot (r_j H_1(att_j)), D'_j = r_j P\}$$

(3) 模拟器 B 将元组 $\{S, R_1, R_2\}$ 添加到一个列表 L_1 当中。

加密询问的流程如下:

(1) 敌手 A 向模拟器 B 选择一个消息明文 M 以及一个访问树 Γ;

(2) 模拟器 B 选择一个秘密数 $s \in Z_p$,然后按照真实的 PB-CP-ABFE 调用快速加密算法和预计算算法,输入公钥 PK、访问树 Γ 以及消息明文 M 并最终产生密文:

$$CT = (\Gamma, \tilde{C} = Me(P_1, P_2)^s, C = sH, \forall y \in Y: C_y = q_y(0)P, C'_y = q_y(0)H_1(att(y)))$$

再次,游戏进入挑战阶段,该阶段的流程如下:

(1) 敌手 A 向模拟器 B 提交 2 个长度相同的消息 M_1 和 M_2;

(2) 模拟器 B 选择一个秘密数 $s \in Z_p$ 以及一个随机的比特 $\theta \in \{0,1\}$ 并返回如下的挑战密文给敌手 A:

$$CT^* = (\Gamma, \tilde{C} = M_\theta Z^s, C = sH, \forall y \in Y: C_y = q_y(0)P, C'_y = q_y(0)H_1(att(y)))$$

然后,游戏再次进入询问阶段。在本阶段,敌手 A 继续向模拟器 B 发送有限次数的私钥询问或加密询问。在敌手 A 发送询问的过程中始终存在如下的限制:

(1) 所有在私钥询问包含的属性集合 S 必须满足 $\Gamma(S) \neq 1$;

(2) 不能将元组 (Γ^*, M_1) 或 (Γ^*, M_2) 作为加密询问的输入。

最后,游戏进入最终猜测阶段,该阶段的流程如下:

(1) 敌手 A 输出 $\theta' \in \{0,1\}$ 作为对 θ 值的猜测;

(2) 模拟器 B 判断 θ' 值,如果 $\theta' = \theta$ 那么敌手 A 赢得挑战游戏,同时模拟器 B 输出 1 表示其认为挑战者 C 产生的 DBDH 难题答案为 $Z = e(P, P)^{abc}$;

(3) 如果 $\theta' \neq \theta$ 那么敌手 A 输掉挑战游戏,同时模拟器 B 输出 0 表示其认为 DBDH 难题答案为 $Z = e(P, P)^z$。

当 $Z = e(P,P)^{abc}$,模拟器产生的挑战密文 CT^* 与真实的 AKE-CP-ABFE 密文服从相同的分布。假设敌手 A 在真实的 AKE-CP-ABFE 算法中区分密文来自于哪段消息明文的优势为 ε',那么模拟器 B 输出 1 的概率为

$$\begin{aligned}&\Pr[B(P, P_1, P_2, P_3, Z = e(g,g)^{abc}) = 1]\\ &= \Pr[\theta = 1 \mid Z = e(P,P)^{abc}] \cdot \Pr[Z = e(P,P)^{abc}]\\ &= (\frac{1}{2} + \varepsilon') \cdot \frac{1}{2} = \frac{1}{4} + \frac{\varepsilon'}{2}\end{aligned}$$

当 $Z = e(P,P)^z$,敌手 A 获取的挑战密文 CT^*

是完全随机分布的,所以敌手 A 实际上只能随机地输出 θ' 值。于是模拟器 B 输出 1 的概率是

$$\begin{aligned}&\Pr[B(P,P_1,P_2,P_3,Z=e(g,g)^z)=1]\\&=\Pr[\theta=1\mid Z=e(P,P)^z]\\&\quad\cdot\Pr[Z=e(P,P)^z]\\&=\frac{1}{2}\cdot\frac{1}{2}=\frac{1}{4}\end{aligned}$$

于是模拟器 B 成功解出挑战者 C 提出的 DBDH 难题的优势满足:

$$\begin{aligned}\varepsilon&\leqslant\left|\begin{array}{l}\Pr[B(P,P_1,P_2,P_3,Z=e(P,P)^{abc})=1]\\-\Pr[B(P,P_1,P_2,P_3,Z=e(P,P)^z)=1]\end{array}\right|\\&=\left|\left(\frac{1}{4}+\frac{\varepsilon'}{2}\right)-\frac{1}{4}\right|=\frac{\varepsilon'}{2}\end{aligned}$$

因此可以断定,如果 DBDH 难题假设成立,即不存在多项式时间算法能够以不可忽略的优势 ε 解决 DBDH 问题,那么也一定不存在多项式时间敌手 A 能够以不可忽略的优势 ε' 区分出密文来自于哪一段消息明文。

4.2　性能分析

本文针对几种典型的 SDN 访问控制系统的功能进行了分析比较,比较结果如表 2 所示。

表 2　SDN 访问控制方案功能比较

方案	数据加密	策略定制	策略粒度	快速加密	抗密钥暴露
文献[7]	否	固定	粗粒度	否	否
文献[8]	否	固定	粗粒度	否	否
文献[19]	是	灵活	细粒度	否	否
AKE-CP-ABFE	是	灵活	细粒度	是	是

在文献[7]的方案里,为了将 SDN 的各个组件在逻辑上进行隔离,事先对不同组件的访问权限进行了详细的描述。文献[8]则是对不同网络组件的可执行指令做出了不同程度的约束,从而实现安全快速的访问控制。以上 2 种方案不仅对 SDN 控制器当中的流表、路由等信息进行加密,还不支持访问策略的灵活制定,因此在不同程度上限制了访问控制方案的安全性和可扩展性。文献[19]基于等级 CP-ABE 设计了一种 SDN 访问控制方案,使得 SDN 控制器能够灵活管理 SDN 的用户、设备和流表数据。相比文献[7]和文献[8]的方案,该方案在安全性和可扩展性上有所提升,同时支持细粒度的访问策略。但是考虑到等级 CP-ABE 庞大的计算量,并不能很好地兼顾整个方案的计算效率。在本文设计的基于 AKE-CP-ABFE 的 SDN 域间信息访问控制系统,在现有的 ABE 算法优势基础上利用预计算技术实现了快速加密。除此之外,即使 SDN 用户或者设备暴露了私钥,其他非法用户也无法通过该私钥获取任何有用的消息明文,可以有效防止私钥暴露产生的信息泄露风险。因此在访问控制的功能上,本文提出的方案具有显著的优势。

为进一步验证基于 AKE-CP-ABFE 的 SDN 跨间访问控制模型的性能,进行了一系列加解密仿真实验。本系统的仿真硬件为 Intel (R) Core(TM) i7-5600U@2.6 GHz,内存为 8 G,系统为 Cent OS 6.7,使用代码库为 JPBC 1.2.1,实验基于 256 位的椭圆曲线,曲线阶为 120 bit 的大素数。实验分别基于文献[12]、文献[18]以及 AKE-CP-ABFE 构建了 SDN 域间访问控制模型,并在不同的属性数量条件下记录了 20 次加解密所需的平均时间。

基于以上 3 种算法构建的 SDN 域间访问控制模型在不同属性数量情况下的平均加密时间如图 2 所示。可以看出,基于文献[18]算法的方案的加密时间要高于基于文献[12]算法的方案,这是因为文献[18]的方案构建了一种重加密机制,使得方案在执行解密的过程中可以将庞大的解密计算部分外包给第三方,同时方便解密者更新属性集合。而在相同属性数量条件下,基于 AKE-CP-ABFE 的 SDN 访问控制方案的平均解密时间约为基于文献[12]方案的 50%,即只需要一般的算力就可以完成相同的加密计算。因此本文提出的模型具备较高的加密效率。

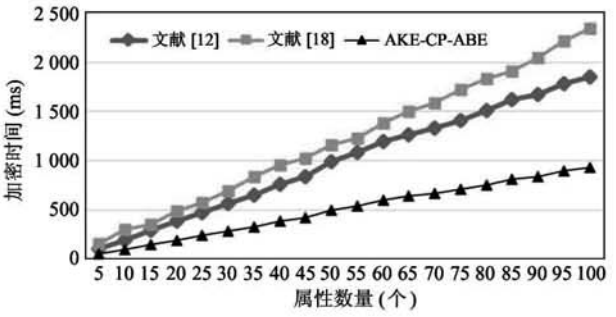

图 2　平均加密时间对比

图 3 展示了基于 3 种方案构建的 SDN 域间访问控制模型在不同属性数量情况下的平均解密时间。可以看出在不同属性数量情况下,文献[18]的方案所需的解密时间都是最多的,而且随着属性数量增加,与文献[12]方案以及 AKE-CP-ABFE 方案的解密时间差越大。这是因为属性越多,额外关于重加密以及属性更新的计算操作就越复杂。而基于 AKE-CP-ABFE 算法构建的 SDN 访问控制模型在解密时间上与文献[12]方案相当,但在解密过程中需要进行 MAC 地址验证所以要多消耗平均 9 ms 的计算时间,总体来说几乎不产生过多的计算量。因此本文提出的方案在解密过程中仍能保持较好的计算效率。

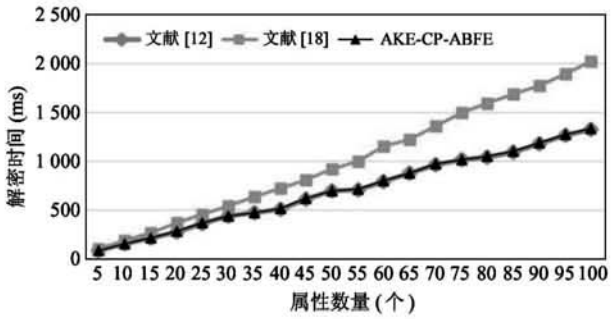

图 3 平均解密时间对比

5 结 论

随着计算机网络规模的扩大,网络管理的难度与日俱增,大型复杂网络呈现爆发式增长。SDN 作为一种控制与数据分离的新型网络架构,为大型网络提供了全新的集中化管理与配置方法。与此同时如何提供安全的信息保护以及灵活的访问控制机制以保证用户及设备的隐私,正成为部署 SDN 尤其是多控制器设置下的大型 SDN 的关键问题。本文提出了一种抗密钥暴露快速属性加密算法,利用预计算技术使得加密者在进行加密时无需进行任何复杂运算,提高了加密效率。同时基于拓展图技术进一步降低了预计算的存储开销。除此之外,将用户或者设备的 MAC 地址嵌入到私钥当中,即使将私钥有意或者无意地暴露给其他非法用户或者设备也不会泄露任何敏感信息。理论分析和仿真实验表明,基于该算法构建的 SDN 域间信息访问控制系统模型,降低了开源 API 造成的用户或者网络设备敏感信息泄露的风险,同时能以较高的计算效率保证敏感信息域间分享的安全性和灵活性。如何在属性动态变化的环境中保证 SDN 访问控制的有效性将是下一步的研究方向。

参考文献

[1] McKeown N, Anderson T, Balakrishnan H, et al. OpenFlow: enabling innovation in campus networks[J]. *Computer Communication Review*, 2008, 38(2): 69-74

[2] Yang L, Dantu R, Anderson T, et al. Forwarding and control element separation (ForCES) framework[R]. Reston: Internet Society, 2004

[3] Feghali A, Kilany R, Chamoun M. SDN security problems and solutions analysis[C]//Proceedings of International Conference on Protocol Engineering and International Conference on New Technologies of Distributed Systems, Paris, France, 2015: 1-5

[4] da Silva E G, Knob L A D, Wickboldt J A, et al. Capitalizing on SDN-based SCADA systems: an anti-eavesdropping case-study[C]//Proceedings of IFIP/IEEE International Symposium on Integrated Net-work Management, Ottawa, Canada, 2015: 165-173

[5] Casado M, Freedman M J, Pettit J, et al. Ethane: taking control of the enterprise[C]//Proceedings of the ACM SIGCOMM 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Kyoto, Japan, 2007: 1-12

[6] Benton K L, Camp J, Small C. Openflow vulnerability assessment[C]//Proceedings of the 2nd ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, Hong Kong, China, 2013: 151-152

[7] Klaedtke F, Karame G O, Bifulco R, et al. Access control for SDN controllers[C]//Proceedings of the 3rd ACM SIGCOMM Workshop Hot Topics Software Defined Networking, Chicago, USA, 2014: 1325-1335

[8] Jäger B, Röpke C, Adam I, et al. Multi-layer access control for SDN-based telco clouds[C]//Proceedings of 20th Nordic Conference on Secure IT Systems, Stockholm, Sweden, 2015: 197-204

[9] Kamath A V, Sudarshan S, Kataoka K, et al. SAFE: software-defined authentication framework[C]//Proceedings of the 12th Asian Internet Engineering Conference, Bangkok, Thailand, 2016: 57-63

[10] Sahai A, Waters B. Fuzzy identity-based encryption[C]//Proceedings of 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques,

Aarhus, Denmark, 2005: 457-473

[11] Goyal V, Pandy O, Sahai A, et al. Attribute-based encryption for fine-grained access control of encrypted data [C]//Proceedings of the 13th ACM Conference on Computer and Communications Security, Alexandria, USA, 2006: 89-98

[12] Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption[C]//Proceedings of 2007 IEEE Symposium on Security and Privacy, Oakland, USA, 2007: 321-334

[13] Chase M, Chow S S M. Improving privacy and security in multi-authority attribute-based encryption[C]//Proceedings of the 2009 ACM Conference on Computer and Communications Security, Chicago, USA, 2009: 121-130

[14] Hur J. Improving security and efficiency in attribute-based data sharing[J]. *IEEE Transactions on Knowledge & Data Engineering*, 2013, 25(10): 2271-2282

[15] 张星, 文子龙, 沈晴霓,等. 可追责并解决密钥托管问题的属性基加密方案[J]. 计算机研究与发展, 2015, 52(10): 2293-2303

[16] 闫玺玺, 孟慧. 支持直接撤销的密文策略属性基加密方案[J]. 通信学报, 2016, 37(5): 44-50

[17] Karati A, Amin R, Biswas G P. Provably secure threshold-based ABE scheme without bilinear map[J]. *Arabian Journal for Science & Engineering*, 2016, 41(8): 1-13

[18] Lin G, Hong H, Sun Z. A collaborative key management protocol in ciphertext policy attribute-based encryption for cloud data sharing[J]. *IEEE Access*, 2017, 5: 9464-9475

[19] He S, Liu J, Mao J, et al. Hierarchical solution for access control and authentication in software defined networks[C]//Proceedings of 8th International Conference on Network and System Security, Xi'an, China, 2014: 70-81

[20] Touati L, Challal Y, Bouabdallah A. C-CP-ABE: cooperative ciphertext policy attribute-based encryption for the Internet of Things[C]//Proceedings of 2014 International Conference on Advanced Networking Distributed Systems and Applications, Bejaia, Algeria, 2014: 64-69

An inter-domain access control scheme for software defined network based on attribute-based encryption

Zhou Bo*, Wang Shulei**

(*School of Electrical of Engineering, Nanjing Vocational College of Information Technology, Nanjing 210023)

(**School of Civil Aviation Flight, Changzhou Institute of Technology, Changzhou 213032)

Abstract

Different from traditional computer network, software defined network (SDN) separates the network control from data transmission. SDN facilitates reducing hardware cost and centralized configuration and management of devices of different vendors. Despite its overwhelming advantages that traditional network architecture does not has, the applications of SDN brings out a series of new problems in terms of security. One of the key problems is how to prevent eavesdropping sensitive information possessed by SDN controllers. However, existing SDN access control schemes cannot provide secure information protection. Furthermore, most schemes focus on single-domain management but fail to meet inter-domain requirements. To address it, an anti-key-exposure ciphertext policy attribute-based fast encryption (AKE-CP-ABFE) is proposed. By introducing pre-computation technique, the encryptors do not need to execute any complicated exponentiation or bilinear map. Besides, MAC addresses are inserted into the private keys so that others cannot use private key of someone. Theoretical analysis proves that AKE-CP-ABFE has great security. Based on the proposed scheme, an inter-domain access control system model with multi-controller setting is established. Simulation experiment demonstrates that it guarantees security and flexibility of inter-domain sensitive information sharing in SDN.

Key words: software defined network (SDN), inter-domain access control, attribute-based encryption (ABE), pre-computation, anti-key-exposure