Capítulo 4: El modelo alternativo Kaa

Con el objetivo de dar una alternativa para evitar o disminuir las limitaciones del modelo base, que se mencionaron en el capítulo anterior, proponemos a Kaa, un nuevo modelo basado en la propuesta de Miklau y Suciu [1] que retoma con un poco más de profundidad la idea que expuso Gifford en [2], acerca de proteger diferentes características o propiedades de un mismo objeto bajo diferentes llaves.

En la sección 4.1 ofrecemos la idea general detrás de la separación de capacidades que proponemos como eje de la solución a las limitaciones del modelo base, comentando, como inicio, las ideas de Gifford que nos inspiraron en la construcción de nuestro modelo. En la sección 4.2 exponemos la manera en la que, apoyándonos de nuevos meta-nodos, reestructuramos cada elemento del árbol de documento para materializar la propuesta de la sección 4.1.

4.1 Separación de funciones de un elemento

Gifford propuso en [2] que un objeto podía ser manipulado de diversas maneras para lograr objetivos específicos y que el control sobre el uso del objeto podía darse a través de asociaciones de llaves sobre las propiedades del objeto, asociaciones que llamó "capacidades". Usando sus *capacidades*, Gifford proponía, entre otras muchas aplicaciones, que un objeto cifrado auto-regulara la manera en la que usuarios con diferentes conjuntos de llaves (privilegios) podían utilizarlo. Para comenzar a entender cómo la propuesta de Gifford puede ayudar a sobrepasar las limitaciones del modelo de

Miklau y Suciu es preciso distinguir las diferentes maneras en las que un elemento en un árbol XML puede ser utilizado, puntualmente, dos funciones:

1. Un elemento en XML se puede emplear para contener información.
2. Un elemento puede contener y servir de ruta hacia varios sub-elementos hijo y, por lo tanto, varios subárboles independientes.

Podemos definir, ahora, lo que será una *capacidad* general para el modelo Kaa: la asociación de una o varias *protecciones específicas* a las funciones de un elemento.

Es fácil, dado lo anterior, desprender dos *capacidades* específicas que se pueden tener sobre un elemento, acceder a su información local y utilizarlo como camino hacia alguno de sus descendientes. Se puede argumentar que los sub-elementos de un nodo son también parte de la información local de éste, dado que en muchos casos son componentes imprescindibles de su semántica; sin embargo y debido a que la intención al implementar control de accesos sobre documentos XML es lograr el dominio con la granularidad más alta posible, consideramos inicialmente que la información local de un elemento es sólo aquello que se encuentra dentro de un nodo pero que no pertenece a ninguno de sus descendientes.

Hasta el momento se ha hablado de los hijos de un elemento sin hacer una distinción clara entre los que tienen asociada una *protección específica* y los que son “libres”; a partir de ahora, es preciso hacer clara la separación: por un lado se tienen los hijos libres, a los que se debe poder acceder simplemente con haber satisfecho la *protección específica* del padre y, por el otro, se encuentran los hijos protegidos sobre los que actúa

otra protección además de la del padre. La distinción es importante porque al requerir, para ser accedidos, la satisfacción de la misma *protección específica* que la información local del padre, los hijos libres se pueden considerar efectivamente parte de ésta y pueden compartir el permiso de acceso dentro de la misma *capacidad*. Por el otro lado, los hijos protegidos se distinguen por haber recibido una *protección específica* además de la del padre, por lo que claramente se separan del resto de su contenido y requirieren, entonces, que su acceso sea controlado con una *capacidad* distinta. Con base en lo anterior se puede decir que el control de acceso sobre todo elemento se debe efectuar a través de dos *capacidades*:

1. El acceso a la información local y a los sub-elementos no protegidos y
2. El acceso a los elementos protegidos del nodo.

La manera en la que Miklau y Suciu definieron el control de acceso a los elementos puede aparentar a primera vista que se consideraron las dos capacidades anteriores: al tener cada elemento *i* protegido con la disyunción de su *protección específica* con las de sus descendientes protegidos, se puede decir que cualquiera que tenga privilegios suficientes para acceder a la información local (o elementos libres) del nodo puede, efectivamente, liberar a *i* y obtener la información que necesita sin conocer la información local de los sub-elementos protegidos j_n (porque estos a su vez estarán resguardados bajo sus *protecciones específicas*); al mismo tiempo, cualquiera que tenga derecho a acceder a uno de los sub-elementos protegidos puede liberar al elemento *i*, llegar hasta el sub-elemento j_k y liberarlo también, conociendo entonces la información local de j_k. Sin embargo, por la manera en la que esos autores definieron su modelo, la

capacidad 1 no es totalmente independiente de la 2 ya que, aunque los sub-elementos protegidos se mantienen protegidos[1] si se accede a la información local de su elemento padre, una vez adquirida la capacidad de acceso a un sub-elemento protegido se adquiere automáticamente el acceso a la información local del padre y eso da origen a la deficiencia discutida en la sección 3.2.2.

El modelo alternativo propuesto intenta mantener las dos capacidades sobre un elemento completamente independientes, de manera que se elimine la relación jerárquica de grados de exposición entre nodos y ancestros y al mismo tiempo se disminuya la capacidad de un atacante de discernir información acerca de los datos protegidos (expuesta en la sección 3.1).

4.2 Agrupación de elementos semejantes

Para lograr la independencia de *capacidades* sobre un elemento, hemos elegido redistribuir sus componentes y emplear meta-nodos para agrupar las características de los elementos de acuerdo a las capacidades a las que se subordinarán. Cabe aclarar que, a pesar de que Miklau y Suciu consideraron el empleo de meta-nodos para posibilitar el trabajo con protecciones, y hasta llegaron a comentar superficialmente en [1] la posibilidad de emplear un meta-padre protegido común para solucionar el problema de indiscreción en el número de hijos protegidos, no valoraron las posibilidades de ese tipo de reestructuraciones y decidieron limitar el uso de meta-nodos a la reescritura de reglas

[1] Aunque, como se comentó en la sección 3.1, se genera la indiscreción respecto al número de sub-elementos protegidos que existen en cada elemento.

de protección, dejando la estructura del árbol original relativamente intacta, como se presentó en el capítulo 2.

En el modelo alternativo Kaa se reconsidera la utilidad de los meta-nodos y la reestructuración, la estructura interna de cada elemento se modifica para agrupar tres diferentes clases de componentes, los sub-elementos protegidos, los sub-elementos libres y la información local. De esta manera es posible construir caminos hacia elementos protegidos ubicados a cualquier altura del árbol, a través de los meta-nodos añadidos, sin revelar en ningún momento información local de algún ancestro del nodo proyectado y, al mismo tiempo, se puede atenuar la indiscreción del número de hijos protegidos presentes cuando se accede a la información local de un elemento.

Para comprender la reestructuración propuesta se puede considerar el ejemplo ilustrado en la figura 6, donde se presenta un elemento *A* cualquiera con una serie de elementos hijos protegidos {*B, C, D*}, otros no protegidos {*E, F, G*} y la información local de *A* (‘*A.data*’). De acuerdo al modelo Kaa, se añaden dos meta-nodos de reestructuración con fórmulas de protección individuales, uno para agrupar cada conjunto de hijos y se logra, de esta manera, una distinción clara entre los sub-elementos del nodo original, basada en la clasificación de capacidades a las que los sub-elementos se subordinan.

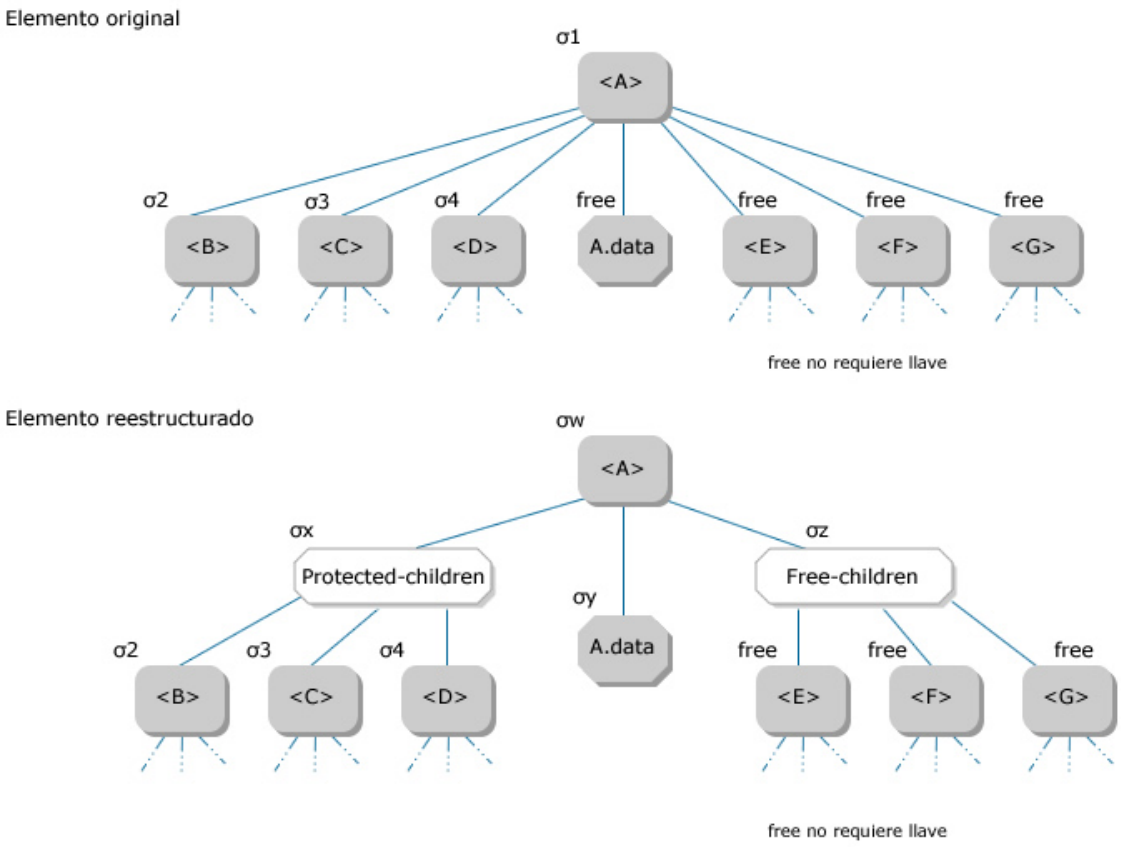

Figura 1: Reestructuración de un elemento de acuerdo a Kaa

Una vez reorganizados y separados en conjuntos los sub-elementos del nodo original, podemos asignar nuevas llaves para controlar el acceso a cada uno de los conjuntos de manera que el acceso a cada uno se preserve independiente de los demás. La construcción de las *fórmulas llave* es, sin duda, la parte fundamental del modelo que proponemos, ya que es mediante estas que podemos mantener inalcanzable el contenido de cada conjunto a menos de que se presenten los privilegios de acceso suficientes para la capacidad indicada. Dado que la construcción de las *fórmulas llave* internas no es trivial, hemos decidido exponerlas en el capítulo 5, en la sección 5.3, una vez que hayamos ofrecido los elementos teóricos necesarios para hacerlas comprensibles.

4.3 Conclusión

La idea general detrás de Kaa, el modelo alternativo que proponemos, es retomar el trabajo de Miklau y Suciu y acercarlo un poco más a la idea original de protección criptográfica de Gifford. Lo anterior se debe a que las limitaciones del modelo base se deben a una deficiente separación de las capacidades inherentes a los elementos del árbol de documento y por lo tanto, es posible mitigarlas replanteando el objetivo de cada sub-elemento del árbol. Hemos propuesto materializar la separación de capacidades reestructurando los componentes al interior de cada elemento y asignando nuevas *fórmulas llave* para los sub-elementos resultantes. En la sección 4.2, hemos mostrado la intuición de cómo logramos esto en el modelo Kaa, sin embargo, el detalle de la estructura de las nuevas fórmulas y otros elementos fundamentales de Kaa hemos preferido exponerlos en el capítulo 5, donde presentaremos a profundidad los conceptos y procedimientos que conforman nuestro modelo.