Análisis de seguridad de un protocolo de intercambio de datos clínicos basado en sistemas multiagente

Albert Brugués de la Torre
Departament d'Enginyeria Telemàtica
Universitat Politècnica de Catalunya
Email: albert.brugues@entel.upc.edu

Magí Lluch-Ariet
Departament d'Enginyeria Telemàtica
Universitat Politècnica de Catalunya
Email: magi.lluch@upc.edu

Josep Pegueroles-Vallés
Departament d'Enginyeria Telemàtica
Universitat Politècnica de Catalunya
Email: josep.pegueroles@upc.edu

***Abstract*—En este artículo se describe la arquitectura de seguridad de *MOSAIC*, un protocolo para el intercambio de datos médicos que soporta acuerdos multilaterales. Los componentes de dicha arquitectura se derivan de una serie de ataques comunes que son susceptibles de ser recibidos por el protocolo. En el artículo se analiza el problema del intercambio justo presente en el protocolo presentando los mensajes de gestión que deben intercambiar los agentes. Debido a los acuerdos multilaterales pueden aparecer bucles durante la fase de negociación del protocolo, así que se presentan los mecanismos utilizados por los agentes para manejar correctamente estos bucles, así como una solución para evitar que agentes malintencionados puedan sacar provecho de esta gestión.**

I. Introducción

La combinación de la medicina con las tecnologías de la información y las comunicaciones (TIC) dan como resultado un gran abanico de nuevas aplicaciones que se enmarcan dentro de lo que se conoce como la telemedicina. Entre estas aplicaciones se encuentran los sistemas de transferencia de registros médicos, cuyo principal objetivo consiste en proporcionar a los profesionales de la salud un conjunto de herramientas que facilitan el acceso a información médica precisa. El uso de estos sistemas comporta toda una serie de ventajas. Por un lado los pacientes reciben un tratamiento de mejor calidad, llegando a ahorrarse segundas visitas [1], por otro el uso de estas tecnologías comporta beneficios económicos [2] derivados entre otros de un mejor aprovechamiento de los recursos.

Si bien el uso de estos sistemas comporta una serie de ventajas, la propia naturaleza de la información a intercambiar está sujeta a normativas éticas [3] y legales [4], [5] que además pueden variar en función del país. Este hecho comporta un gran reto en el diseño de estos sistemas ya que es primordial asegurar la seguridad y la privacidad de la información que se intercambian sobre los pacientes los distintos centros médicos, especialmente cuando las conexiones se realizan a través de redes públicas no seguras como Internet.

En la actualidad existen diferentes soluciones que facilitan el intercambio de registros médicos. Entre ellos se encuentran estándares como DICOM [6] para imágenes médicas, ISO/EN 13606 [7] y HL7 [8] para el intercambio de historia clínica electrónica (HCE), así como diferentes sistemas basados en agentes con propósitos diversos [9].

MOSAIC [10] es un protocolo basado en sistemas multiagente que incluye la seguridad como parte esencial e integrante de su diseño. Su finalidad es facilitar el proceso de intercambio de datos clínicos, y su principal característica reside en el hecho de soportar acuerdos multilaterales entre distintas partes. De este modo y asumiendo la aprobación por parte de los comités de ética, cuando un clínico obtiene datos sobre sus pacientes puede buscar casos similares en otros centros y compararlos con los que ya tiene.

En este artículo se presenta la arquitectura de seguridad de *MOSAIC* y se diseñan algunos de los componentes de la misma. El resto del documento está organizado de la siguiente forma: en el resto de la sección I se resume el funcionamiento del protocolo presentado en [10], [11]. En la sección II se revisan diversos enfoques de seguridad existentes para sistemas médicos. En la sección III se describen las amenazas y ataques que puede recibir el protocolo. En la sección IV se presenta la arquitectura de seguridad del protocolo. En la sección V se analiza con más detalle dos de los componentes de la arquitectura. Finalmente en la sección VI se exponen las conclusiones y el trabajo futuro a realizar.

A. Escenario

El escenario de actuación del protocolo consiste en una serie de nodos interconectados cada uno con un *Data Mart* asociado y una plataforma de agentes. En el *Data Mart* de cada nodo se almacenan casos de tipos de datos clasificados por categorías (p. ej. casos sobre tipos de tumores cerebrales). Un usuario del sistema puede decidir que tipo de datos de su *Data Mart* desea compartir con el resto de nodos de la red, o puede requerir el acceso a un tipo de datos compartido por el resto de nodos.

El proceso para entregar al usuario el tipo de datos que ha solicitado se puede dividir en cuatro fases:

1) Exploración de la red: empieza cuando el usuario selecciona un tipo de dato de su interés. Este recurso puede encontrarse en distintos nodos, así que para cada opción se exploran los distintos caminos de negociación que permiten acceder al recurso.

2) Selección de los acuerdos: para cada recurso se selecciona un camino entre todos los encontrados. La notificación del camino seleccionado se transmite a todos los agentes que forman parte del acuerdo.
3) Transmisión de datos: después de recibir la notificación del camino seleccionado los agentes proceden al intercambio de los datos.
4) Finalización de la transacción: en el caso de que todos los nodos participantes hayan recibido los datos correctamente se transmite la autorización para su uso, o se manda una revocación en caso contrario

Un ejemplo simplificado de la fase de exploración es el siguiente:

- Un usuario de un nodo A hace una petición a un nodo B para acceder a los casos que comparte sobre un determinado tipo de datos.
- El nodo B puede ofrecer a A el acceso a sus datos de manera libre o puede poner como restricción de acceso el poder acceder a los casos de un determinado tipo de datos.
- Si A dispone de los datos exigidos por B, entonces A permite a B el acceso a esos datos y B permite el acceso a los datos requeridos por A. Produciéndose así un posible acuerdo bilateral.
- Si A no puede cumplir la restricción de acceso impuesta por B, entonces deberá buscar un nodo C que permita a B el acceso a esos datos. Entonces C puede exigir a A como restricción de acceso el poder acceder a un determinado tipo de datos. De este modo empieza un proceso recursivo que de solucionarse acaba en un posible acuerdo multilateral entre distintos nodos.

El proceso de exploración de un camino termina con un posible acuerdo cuando el nodo que recibe la petición cede sus datos de forma libre, o cuando impone como restricción de acceso la cesión de un tipo de datos que tenga el nodo que ha realizado la petición.

Al finalizar el proceso de exploración, el nodo que inició la petición escoge uno entre todos los acuerdos encontrados. La elección de un acuerdo determinado implica la transferencia de los datos acordados durante el proceso de exploración. Cuando un nodo ha recibido los datos acordados envía su confirmación al nodo que inició la petición. Al recibir todas las confirmaciones el nodo inicial envía la autorización de uso de los datos a cada nodo. Al finalizar el proceso todos los nodos participantes en el acuerdo de negociación escogido han conseguido tipos de casos de su interés. Estos casos conseguidos pueden ser usados por los usuarios del sistema para compararlos con los que ya tienen y poder realizar mejores diagnósticos.

B. Agentes del protocolo

El proceso de negociación para el intercambio de los datos está automatizado mediante el uso de los siguientes agentes:

- *Multicast Contributor* (MCC): Lo activa un usuario para ofrecer un tipo de datos a los usuarios del resto de nodos, con o sin restricciones de acceso.
- *Unicast Contributor* (UCC): Lo activa el MCC para procesar la petición recibida por parte de un agente Multicast Petitioner.
- *Multicast Petitioner* (MCP): Este agente lo activa el usuario para explorar la red en busca de un determinado recurso. También puede ser activado por un agente Unicast Petitioner para resolver una restricción de acceso de un UCC cuando el tipo de datos exigidos no se encuentra en el propio nodo del UCP.
- *Unicast Petitioner* (UCP): Lo activa un MCP para negociar con un UCC el acceso a un determinado tipo de datos.
- *Yellow Pages* (YP): Este agente actúa como un servicio de directorio, donde se registran los MCC activos en la red.

Por simplificación en el resto del artículo se referirá como *Contributor* la unidad formada por un agente MCC y uno de sus agentes UCC, y como *Petitioner* la unidad formada por un agente MCP y uno de sus agentes UCP.

El lector interesado encontrará en [10] una descripción más detallada de los mismos.

II. Seguridad en sistemas de intercambio de registros médicos

Un sistema de intercambio de registros médicos como *MOSAIC* debe cumplir como mínimo los requisitos de seguridad básicos de los sistemas de información en general, y además se deben de complementar con sus necesidades específicas.

Para definir los requisitos generales existen toda una serie de estándares y recomendaciones internacionales que pueden servir como guía.

En primer lugar los Criterios Comunes, estandarizado como ISO/IEC 15408 [12], es un *framework* que ofrece una taxonomía para evaluar la funcionalidad de la seguridad a través de un conjunto de requerimientos funcionales y de garantía. Entre ellos se encuentran: comunicación/no repudio, soporte criptográfico, protección de los datos de usuario, identificación y autenticación, y privacidad.

El estándar FIPS-200 del *National Institute of Standards and Technology* [13] especifica los mínimos requerimientos de seguridad en 17 áreas en relación a la protección de la confidencialidad, integridad, y disponibilidad en sistemas de información federados así como de la información procesada, almacenada y transmitida por estos sistemas.

Además de estos estándares, varios autores han publicado arquitecturas específicas para sistemas de intercambio de registros médicos. Estas arquitecturas están centradas en el acceso seguro a la HCE utilizando diferentes aproximaciones. Alzharani et al. [14] proponen un modelo de dos niveles. El primer nivel consiste en el diseño del software que incluye los aspectos formales de seguridad como mecanismos, algoritmos, datos y servicios. El segundo nivel es el dominio de aplicación que incluye factores como autenticación, control de acceso, acceso a los datos y validación de los datos. Otra arquitectura distinta es la de Gritzalis y Lambrinoudakis [15] que se basa en

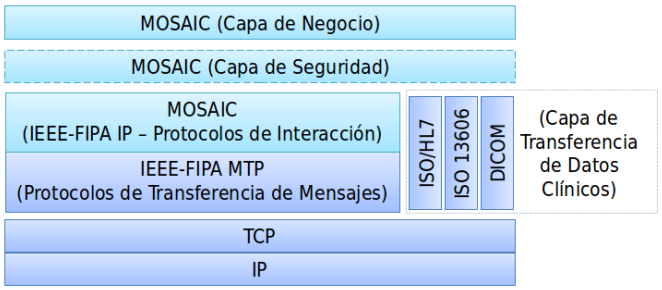

Figura 1. Pila de protocolos donde se ubica *MOSAIC*

emplear, para cada centro de salud, un agente responsable de la seguridad que se ocupa de la autenticación de los usuarios y del control de acceso a la información en función del rol del usuario. Los requisitos de seguridad que se satisfacen con esta arquitectura son la confidencialidad de los datos intercambiados, la integridad y el control de acceso del contenido, servicios de autenticación de usuario de tipo *single sign-on*, autorización apropiada para usuarios remotos de los que se dispone de poca información y el registros de las acciones llevadas a cabo por los usuarios. Otras arquitecturas como la de Chen et al. [16] emplean agentes móviles para acceder a la HCE. La seguridad de este esquema se basa en la capacidad de gestión de claves usando la interpolación polinómica de Lagrange mediante un sistema de control de acceso jerárquico. Esta arquitectura presenta robustez a distintos tipos de ataques externos e internos. Los ataques externos se basan en la obtención de la clave de descifrado a partir de los parámetros públicos del cifrado, mientras que los internos pueden ser ataques inversos o ataques cooperativos ambos basados en la obtención de la clave de descifrado de un nivel jerárquico superior.

Finalmente Kailar y Muralidhar [17] proponen un modelo general de seguridad para estos sistemas. El modelo se basa en una serie de requisitos de seguridad extraídos de los casos de uso de tres sistemas distintos, así como un conjunto de mecanismos de seguridad y políticas de acceso necesarias para cumplir con dichos requisitos. De este modo el modelo consigue hace frente de forma exitosa a una serie de amenazas comunes como son: consumo/producción de datos por parte de usuarios no autorizados, confidencialidad e integridad comprometidas, virus y *spyware*, denegación de servicio y suplantación de identidad. Los mecanismos propuestos están pensados para hacer de la privacidad y la seguridad los objetivos en confidencialidad, integridad y disponibilidad.

Todas estas arquitecturas presentan la característica común que tratan explícitamente la seguridad para garantizar ciertos requerimientos. En *MOSAIC* la arquitectura de seguridad es una capa integrada en el diseño de su arquitectura general (Figura. 1). De este modo los agentes participantes en el protocolo pueden realizar sus funciones específicas junto con unas medidas de seguridad que mejoran su comportamiento normal.

III. Posibles amenazas y ataques

Asumiendo que el protocolo no dispone de ningún mecanismo de seguridad, un primer análisis de las posibles amenazas y ataques concluye que los atacantes externos pueden

1) realizar ataques pasivos, por ejemplo con la monitorización de la actividad de los agentes se puede conseguir información acerca de cuanta información intercambian los nodos y con que frecuencia.
2) realizar ataques activos como
   a) denegación de servicio de un nodo o del servicio de directorio. En el caso de realizar este ataque al servicio de directorio se imposibilitaría la comunicación entre los nodos.
   b) la suplantación de identidad tanto de un nodo determinado como al servicio de directorio. En este caso la impersonización de un nodo podría ser utilizada para la ejecución de agentes malintencionados que se aprovechen de brechas en los agentes legítimos del sistema.
   c) la eliminación de los mensajes que se intercambian los agentes a través de la red que alteraría la interacción entre los agentes.
   d) la alteración de los mensajes por ejemplo cambiando el tipo de dato que el usuario solicita.
   e) fabricación y envío de mensajes con malos propósitos a agentes del sistema, como por ejemplo un mensaje de eliminación del agente.
   f) el acceso no autorizado a los datos almacenados por los nodos que comprometería su integridad y confidencialidad.

Respecto a los usuarios internos hay que tener en cuenta que pueden

1) hacer un mal uso del sistema como por ejemplo
   a) no aceptando las peticiones de acceso a la información que comparte de manera premeditada.
   b) compartiendo en la red información la cual no está autorizado compartir. En este caso tanto puede ser que los derechos de la información a querer compartir pertenezcan a otro usuario del mismo nodo o a un usuario de un nodo externo.
   c) transfiriendo datos que no se corresponden con la descripción dada.
2) realizar ataques activos utilizando agentes malintencionados para beneficios propios.
3) intentar aumentar la reputación de sus agentes o disminuir la reputación de los agentes de otros nodos. En este caso varias opciones son posibles como
   a) intentar modificar el valor de la reputación directamente.
   b) realizar un ataque Sybil creando nodos falsos para aumentar indirectamente la propia reputación.
   c) la confabulación de dos nodos distintos para intentar aumentar mutuamente su reputación.
   d) no actualizar como es debido la reputación del usuario que ha mandado los datos una vez ha

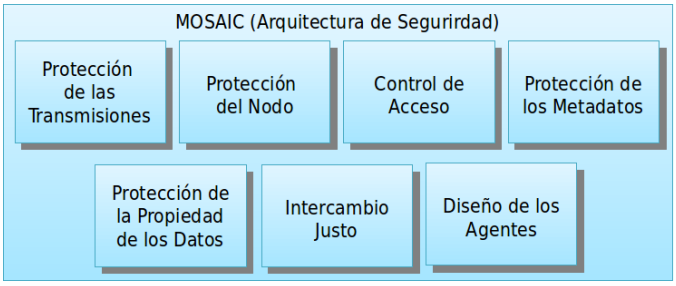

Figura 2. Componentes que forman la arquitectura de seguridad de *MOSAIC*

finalizado satisfactoriamente una transacción de información, produciéndose así una situación de desventaja para este último.

4) alterar el proceso de intercambio de datos
   a) no iniciando la transmisión de los datos cuando es debido.
   b) transmitiendo información aleatoria.
   c) no confirmando la correcta recepción de los mismos.

En todos estos casos se debe garantizar que sólo se pueda acceder y utilizar los datos enviados en el caso de que los datos recibidos sean los acordados durante la fase de exploración.

IV. Arquitectura de seguridad del protocolo

En base a los ataques descritos anteriormente la arquitectura de seguridad de *MOSAIC* (Figura. 2) consta de los siguientes bloques:

- Protección de las transmisiones: garantiza la integridad y confidencialidad de los datos enviados a través de la red así como la autenticidad de las entidades que se comunican.
- Protección del nodo: rechaza cualquier intento de acceso externo no autorizado a la red interna del nodo.
- Control de acceso: garantiza que el sistema se utiliza correctamente por parte de los usuarios.
- Protección de los metadatos: garantiza que la reputación de los agentes y los nodos no se vea alterada de forma fraudulenta.
- Protección de la propiedad de los datos: tiene como finalidad garantizar que ningún usuario atenta contra la propiedad de los datos.
- Intercambio justo: asegura que en todo intercambio de datos ambas partes reciben los datos esperados y ofrece el servicio de no repudio a ambas partes.
- Diseño de los agentes: define los comportamientos que deben tener los agentes ante la recepción de mensajes no esperados.

El módulo de protección de las transmisiones garantiza la privacidad e integridad de las comunicaciones en los mensajes que intercambian los agentes del protocolo. Este objetivo se alcanza mediante el cifrado y uso de funciones hash en los mismos. Esto se aplica tanto a comunicaciones entre agentes de distintos centros como a comunicaciones entre agentes de un mismo centro, ya que es posible que agentes de un mismo centro requieran comunicación y actúen desde distintas máquinas. Además este módulo es responsable de proporcionar la identidad del nodo al resto de nodos y garantizar la identidad de los nodos con los que se establecen comunicaciones.

El módulo de protección del nodo es el responsable de proteger al nodo frente a cualquier intento de acceso no autorizado a su red interna. Especialmente se protege el acceso a la base de datos donde se almacenan los datos compartidos ya que un acceso no autorizado a estos datos comprometería su confidencialidad e integridad. Para este propósito cada nodo dispone de su cortafuegos y sistema de detección de intrusos.

El control de acceso determina en cada nodo que usuarios pueden hacer uso del sistema y define las acciones que éstos pueden realizar. Por ejemplo puede ser que un usuario este autorizado a realizar peticiones de intercambio de información con otros nodos pero no a compartir cierta información en la red, o que un usuario no este autorizado a confirmar una petición de intercambio de datos.

El módulo de protección de los metadatos tiene la finalidad de proteger la reputación de los agentes de cada nodo. Sus objetivos son que la reputación no se vea alterada de forma fraudulenta y que se actualiza como es debido. Este módulo es esencial en la arquitectura de seguridad de *MOSAIC* ya que la reputación de los agentes es un parámetro a tener en cuenta en la decisión de un determinado camino de negociación.

El módulo de protección de la propiedad de los datos tiene el propósito de detectar si los datos obtenidos en un intercambio fueron generados en el propio nodo. El hecho de que dos nodos intercambien información no tiene que implicar que ambos estén autorizados a compartirla con otros nodos. Para facilitar el seguimiento de los datos este módulo registra información (fecha, nodo destinatario, usuario que izo la petición, etc) de cada transacción completada satisfactoriamente e introduce técnicas de fingerprinting a los datos.

El intercambio justo es un módulo que garantiza que cuando se acepta un acuerdo de negociación, si éste finalmente es seleccionado para la transferencia de datos, los datos que se reciben son los correctos y que sólo en este caso se puede acceder y utilizar los datos enviados. Este módulo proporciona el servicio de no repudio para proteger a los nodos contra comportamientos deshonestos.

El último de los módulos, el diseño de los agentes, está relacionado con el comportamiento que tienen los agentes en función de los mensajes que reciben. Un agente debe responder adecuadamente frente a mensajes inesperados y se debe evitar que su diseño presente brechas que puedan ser aprovechadas por agentes malintencionados.

Finalmente cabe destacar que los casos compartidos por los nodos sólo contendrán aquella información que sea relevante para la práctica médica. Se asume que los datos están totalmente anonimizados de tal modo que su propiedad recae sobre el clínico que los ha generado y decide compartirlos en la red. Esto significa que en los datos compartidos por los nodos se ha destruido el nexo de unión con la persona a la cual pertenecen

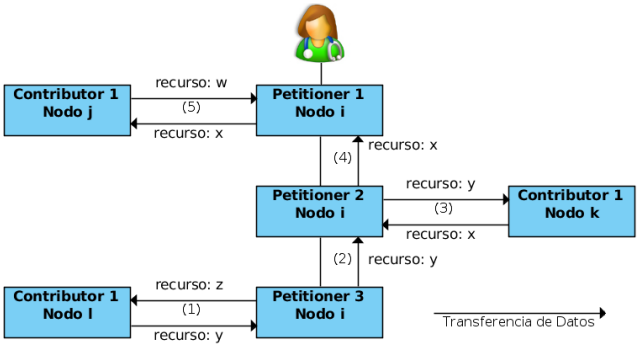

Figura 3. Proceso de transferencia de los datos

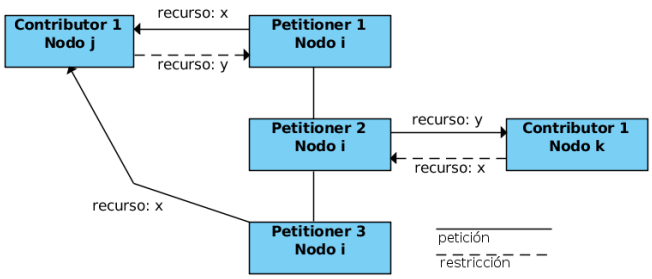

Figura 4. Ejemplo de bucle durante el proceso de negociación

de tal modo que no es posible asociarlos con certeza a una persona determinada.

V. Análisis de seguridad para intercambio justo y comportamientos malintencionados de los nodos

Algunos de los componentes de la arquitectura de seguridad de *MOSAIC* presentan soluciones que han sido muy tratadas en la literatura. En esta sección se exponen los problemas del intercambio justo y un posible comportamiento malintencionado de los nodos.

A. Mensajes de gestión fabricados

Durante la etapa de exploración del protocolo se produce un intercambio de mensajes entre agentes de distintos nodos. Este intercambio sirve para determinar que tipo de recurso tiene que ofrecer el *Petitioner* de un nodo al *Contributor* de otro nodo para que se pueda producir un intercambio de datos. Cuando en el nodo del *Petitioner* no hay ningún *Contributor* que ofrezca el recurso demandado, entonces el *Petitioner* crea un nuevo *Petitioner* para que se lo busque y así poder ofrecérselo al *Contributor* que se lo ha pedido. Al final del proceso pueden existir muchos caminos distintos para conseguir el recurso de interés del usuario y se debe escoger uno para que los agentes inicien la transferencia de los datos. El *Petitioner* inicial es el encargado de escoger este camino y notificar al resto de *Petitioners* que se inicie la transferencia. En la Figura 3 se muestra un ejemplo del proceso de intercambio de datos. En este ejemplo la fase de exploración termina cuando al nodo i se le exige un recurso el cual dispone, pero esta fase también podría terminar si un *Contributor* ofreciera al nodo i los datos de forma libre. Como se puede observar todos los *Petitioners* que se desencadenan en una rama de negociación (a excepción del creado por el usuario) desempeñan una función de intermediarios entre los *Contributors* que participan en el acuerdo.

El proceso de intercambio de datos puede terminar con una transacción completa y satisfactoria por parte de todos los agentes o puede terminar con un error en alguno de los nodos. Una transacción se considera satisfactoria cuando todos los *Contributors* y el *Petitioner* inicial hayan recibido los datos acordados durante la fase de exploración. En este caso cada *Contributor* envía un mensaje ACK que se hace llegar al *Petitioner* inicial o un mensaje NACK en caso contrario. Después de recibir todos los ACK el *Petitioner* inicial manda un mensaje de COMMIT al resto de agentes confirmando la autorización de uso de los datos recibidos. En el caso de que no reciba todos los ACK o reciba un NACK, manda un mensaje de ROLLBACK al resto de agentes desautorizando el uso de los datos recibidos.

En este proceso pueden existir comportamientos que alteren el funcionamiento normal del protocolo. Por ejemplo puede pasar que un *Contributor* mande un NACK en vez de un ACK, o que un *Petitioner* envíe ROLLBACK en vez de COMMIT. Estos problemas se pueden solucionar introduciendo en el protocolo mecanismos de no repudio. Estos mecanismos pueden utilizarse durante la fase de transmisión de datos según el grado de confianza entre los nodos involucrados en el intercambio. Además la reputación de los agentes juega un papel fundamental a la hora de escoger un determinado camino. Haciendo uso de esta información se puede evitar la interacción con agentes que hayan tenido estos comportamientos en el pasado. Es por ello que cada vez que una transacción de intercambio de datos se completa satisfactoriamente los usuarios que hayan participado en ella tienen la opción de aumentar el valor de la reputación de los agentes de quien hayan recibido datos. De este modo cuando el *Petitioner* tiene que seleccionar un camino, es más deseable seleccionar aquellos caminos donde no hayan *Contributors* con baja reputación o que en el pasado hayan presentado un comportamiento inesperado. Otro ejemplo distinto donde puede ser útil la reputación es durante la fase de exploración del protocolo: cuando un *Petitioner* pide los datos a un *Contributor* éste puede rechazar directamente la petición en función de la reputación del destinatario final de los datos. Por otro lado la reputación manifiesta el grado de colaboración de los agentes de modo que aquellos agentes que no fomentan la colaboración a la larga salen perjudicados.

B. Aprovechamiento malicioso del bucle en la fase de negociación

Durante la fase de negociación del protocolo existe la posibilidad de que se produzca un bucle entre dos nodos. Un bucle ocurre cuando un *Petitioner* de un nodo realiza una petición a un *Contributor* de otro nodo el cual ya había recibido otra petición de otro *Petitioner* perteneciente a la misma rama de negociación. Como se puede ver en la Figura 4

este caso especial se da cuando un *Contributor* exige como restricción a un *Petitioner* un tipo de recurso que ya había sido solicitado anteriormente por otro *Petitioner*.

El comportamiento que tiene un *Contributor* frente a esta situación es no imponer ninguna restricción de acceso al recurso solicitado por el *Petitioner* que le hace la segunda petición. De este modo al ceder los datos de manera libre al segundo *Petitioner* todos los nodos participes en esa cadena de negociación salen beneficiados.

La realización de este proceso implica que tanto el *Petitioner* como el *Contributor* deben implementar funcionalidad extra para poder gestionar esta situación correctamente. Esta gestión se realiza mediante el procesamiento de un paquete *Request* por parte de ambos. Este paquete es un objeto que identifica la petición y que consta de los siguientes campos:

- El identificador del nodo que realiza la petición
- El identificador del primer agente MCP de la cadena de negociación, es decir el activado por el usuario
- El identificador de la rama de negociación

Cuando un usuario crea un *Petitioner* para buscar un determinado tipo de recurso el agente a su vez genera un paquete *Request*, rellenando los dos primeros identificadores de la lista con sus valores correspondientes y genera un entero aleatorio para el tercer identificador. A medida que los *Petitioners* van explorando las diferentes opciones de negociación existentes se va construyendo un árbol de agentes donde cada rama representa un camino de negociación. Cada nuevo agente *Petitioner* creado recibe el *Request* de su agente creador al que le añade un nuevo entero al identificador de la rama de negociación. De este modo cuando termina el proceso de se tiene un *array* que posibilita la identificación de cada una de las distintas ramas que se han creado. A continuación este nuevo *Request* generado se manda con la petición de acceso que realiza el *Petitioner* al *Contributor*. El *Contributor* por su parte guarda en una tabla todos los *Request* activos recibidos hasta el momento, y cada vez que recibe uno nuevo lo compara con los que ya había recibido para determinar si cede los datos de manera libre o no.

Asumiendo un comportamiento bien intencionado de los agentes la estructura que tiene el paquete *Request* proporciona seguridad en los bucles que se pueden dar durante el proceso de negociación. La función de los dos primeros identificadores es evitar que dos *Petitioners* de distintos nodos, o de un mismo nodo pero de árboles de negociación distintos, puedan realizar una petición a un mismo *Contributor* y éste interprete que la segunda petición pertenece a la misma rama que la primera. Por otro lado el uso de enteros aleatorios para la identificación de las ramas también proporciona seguridad extra. Como los identificadores de los agentes son únicos en cada nodo estos también se podrían usar para identificar las ramas del árbol, pero de este modo con el uso de enteros aleatorios los *Contributors* no reciben información sobre los identificadores de los *Petitioners* del nodo con el que están negociando.

En este esquema presentado existen dos vulnerabilidades de tipo distinto que podrían ser aprovechadas por agentes maliciosos. El objetivo del diseño de los agentes es poder

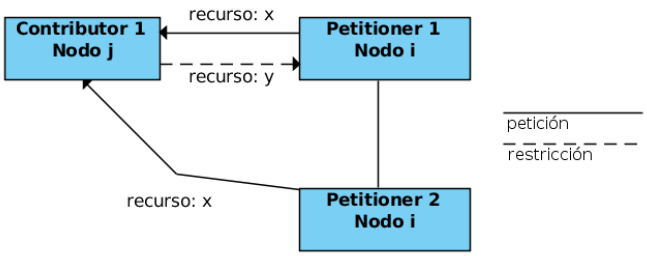

Figura 5. Ejemplo de una petición ilícita por parte de un *Petitioner*

detectar estos comportamientos durante la fase de exploración del protocolo. En las Figuras 5 y 6 se muestran los dos ejemplos.

1) Ataque directo: En el caso de la Figura 5 el *Petitioner* de un nodo realiza directamente una segunda petición mediante la modificación apropiada del identificador de la rama en el paquete *Request*. Con esta segunda petición el *Contributor* cedería los datos de forma libre en la fase de intercambio de datos sin recibir nada a cambio. Debido a este problema es esencial que en este caso especial el *Petitioner* informe al *Contributor* quién es el destinatario final de los datos, ya que de ser el mismo nodo en ambas peticiones se rechazaría directamente la petición. Teniendo en cuenta esta consideración un *Petitioner* que quiera realizar un ataque directo para aprovecharse del funcionamiento de los bucles, debe informar como destinatario final de los datos un *Contributor* que comparta un recurso del tipo exigido por el *Contributor* que es víctima del ataque. Este recurso puede ser ofrecido de forma libre o mediante la entrega de un recurso de tipo distinto en cuyo caso no se altera el funcionamiento normal del protocolo.

Aún así queda la posibilidad que un *Petitioner* y un *Contributor* de dos nodos distintos confabulen para realizar este ataque. En este caso hay que tener en cuenta que la notificación de la restricción de acceso se realiza una vez recibida la petición, y por tanto se deben cumplir las siguientes condiciones:

- El nodo del *Contributor* atacante debe disponer el tipo de recurso solicitado.
- El *Contributor* atacante tiene que haber sido creado después de la petición del *Petitioner* atacante al *Contributor* atacado.

Este hecho puede ser utilizado por el *Contributor* víctima para detectar que está siendo atacado. El Yellow Pages puede proporcionarle la fecha y hora cuando se dio de alta en la red el *Contributor* atacante.

2) Ataque indirecto: En el caso de la Figura 6, la segunda petición que se realiza al *Contributor* del nodo k es lícita y por tanto a priori no se puede detectar ninguna anomalía en la fase de exploración del protocolo. En este caso se puede hacer un uso no autorizado de los datos (de tipo y) si el *Petitioner* 4 los entrega directamente al *Petitioner* 1 para el *Contributor* del nodo j. Como este *Contributor* no es el destinatario final de los

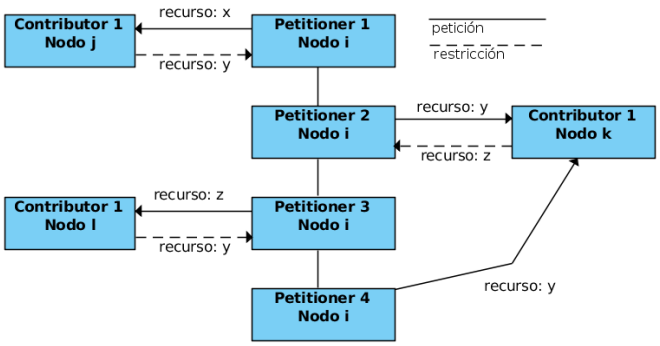

Figura 6. Ejemplo de petición lícita. En este caso el *Petitioner* 4 podría hacer un uso fraudulento si manda el recurso 'y' al *Petitioner* 1

datos, se provoca una situación de desventaja para todos los *Contributors* situados en el medio de la cadena de negociación (los de los nodos k y l en el ejemplo). Es por tanto esencial que en el cifrado realizado a los datos se utilice o bien la clave pública del destinatario final en el caso de utilizar criptografía de clave asimétrica, o bien una clave compartida por el emisor inicial y el receptor final de los datos en el caso de utilizar criptografía de clave simétrica.

VI. Conclusiones y trabajo futuro

En este artículo se han analizado los componentes que deben formar parte de la arquitectura de seguridad de *MOSAIC*, un protocolo para el intercambio seguro de registros médicos. La naturaleza personal de los datos a intercambiar por este protocolo hace que sea esencialmente importante que tenga implementados unos mecanismos de seguridad robustos. En base al análisis de una serie de ataques comunes que puede sufrir el protocolo se han identificado un total 7 bloques distintos que forman la arquitectura de seguridad del protocolo que son los siguientes: protección de las transmisiones, protección del nodo, control de acceso, protección de los metadatos, protección de la propiedad de los datos, intercambio justo y el diseño de los agentes.

Para el módulo de intercambio justo se ha propuesto un esquema de los mensajes de gestión que deben intercambiar los agentes para notificación de la correcta recepción de los datos o fallida de la misma. En el módulo relacionado con el diseño de los agentes se ha presentado la gestión del del bucle en la fase de negociación del protocolo, cómo un agente malintencionado puede aprovecharse de este bucle y se ha propuesto una solución a este problema basada en la notificación del receptor final de los datos.

El trabajo futuro a realizar en la arquitectura de seguridad de *MOSAIC* es:

- Formalizar el protocolo de intercambio justo que deben seguir todos los agentes.
- Implementar un esquema de fingerprinting/watermarking para proteger la propiedad de los datos médicos compartidos.
- Realizar un análisis exhaustivo de los mensajes intercambiados por los agentes para definir una mayor seguridad.
- Finalmente se espera implementar la arquitectura descrita en un escenario real.

Agradecimientos

Este trabajo se ha realizado con el soporte del proyecto TAMESIS (TEC2011-22746), y del AGAUR con la beca 2010-TEM-88.

Referencias

[1] K. Harno, P. Nykanen, J. Ohtonen, A. Seppala, K. Kopra, "Healthcare Information Exchange in Regional eHealth Networks Implications for Initiatives in Advancing Shared Care", *eHealth, Telemedicine, and Social Medicine, 2009. eTELEMED '09. International Conference on* , vol., no., pp.42-45, 1-7 Feb. 2009 doi:10.1109/eTELEMED.2009.24

[2] K.A. Stroetmann, T. Jones, A. Dobrev, V.N. Stroetmann, "eHealth is Worth it. The economic benefits of implemented eHealth solutions at ten European sites", European Commission Information Society and Media Directorate-General: Brussels, 2006.

[3] UNESCO. Universal declaration on bioethics and human rights. Paris. June 2006 en http://unesdoc.unesco.org/images/0014/001461/146180E.pdf (Último acceso: 05-06-2012).

[4] Directiva 95/46/CE del Parlamento Europeo y del Consejo de 24 de Octubre de 1995 relativa a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos.

[5] The Health Insurance Portability and Accountability Act of 1996 (HIPAA).

[6] DICOM. Digital imaging and communication in medicine, 1993.

[7] ISO 13606. Electronic health record communication - part 1: Reference model, 2008.

[8] ISO/HL7 27931. HL7 vesion 3 - Reference information model, 2006.

[9] D. Isern, D. Sánchez and A. Moreno, "Agents applied in health care: A review", International Journal of Medical Informatics, 79(3):145 – 166, 2010.

[10] M. Lluch-Ariet and J. Pegueroles-Vallés, "The mosaic system - a clinical data exchange system with multilateral agreement support", 3rd International ICST Conference on Electronic Healthcare for the 21st century. Casablanca, 12/2010 2010.

[11] M. Lluch-Ariet, A. Brugués de la Torre, J. Pegueroles-Vallés, "Performance evaluation of MOSAIC: A Multi Agent System for multilateral exchange agreements of clinical data", Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AA-MAS 2012), Valencia, June 2012.

[12] ISO/IEC 15408 Standard. Common Criteria for Information Technology Security Evaluation Version 2.3. 2005.

[13] FIPS PUB 200, Minimum Security Requirements for Federal Information and Information Systems, Technical report, National Institute of Standards and Technology, 2006.

[14] S. Al-zharani, S. Sarasvady, H. Chandra, P. Pichappan, "Controlled EHR access in secured health information system", *Digital Information Management, 2006 1st International Conference on* , vol., no., pp.63-68, 6-6 Dec. 2006 doi: 10.1109/ICDIM.2007.369331

[15] D. Gritzalis and C. Lambrinoudakis, "A security architecture for interconnecting health information systems", International Journal of Medical Informatics, Volume 73, Issue 3 , 31 March 2004, Pages 305-309.

[16] T. Chen, Y. Chung, F. Lin, "A Study on Agent-Based Secure Scheme for Electronic Medical Record System", Journal of Medical Systems, pp. 1-13-13 Sep. 2012 doi:10.1007/s10916-010-9595-8

[17] R. Kailar and V. Muralidhar, "A security architecture for health information networks", In *AMIA Annual Symposium Proc.*, pages 379–383, 2007.