UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

AUTENTICACIÓN DESMENTIBLE EN CANALES ANÓNIMOS

PRESENTACIÓN DEL TEMA DE MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

ALONSO EMILIO GONZÁLEZ ULLOA

PROFESOR GUÍA:
ALEJANDRO HEVIA ANGULO

MIEMBROS DE LA COMISIÓN:
GONZALO NAVARRO BADINO
RODRIGO PAREDES MORADELA

SANTIAGO DE CHILE
MARZO 2011

Agradecimientos

Agradezco a mi profesor guía Jo por su apoyo y orientación a lo largo de este trabajo. Nahid Akbar (a.k.a. Killer Civilian) por sus opiniones y conociemientos con *Kaillera p2p*. `iq_132` y los usuarios de los foros de FinalBurn Alpha por su ayuda con el código del emulador. A Joaquín, Sebastián y Haníbal por sus *feedbacks*.

Felipe Lema S.

Índice General

Índice de figuras

Capítulo 1

Introducción

1.1. Motivación

Alicia y Roberto forman un matrimonio de muchos años al igual que sus amigos Marcelo y Beatriz. Con el pasos de los años ambos matrimonios han ido perdiendo la fuerza y la rutina ha ido socabando la pasión que alguna vez existió. Es así como Alicia decidió iniciar una aventura con el marido de su amiga Beatriz, Marcelo, y coincidentemente Beatriz decidio hacer lo mismo con el marido de su amiga.

Hace algún tiempo Alicia comenzó a tomar un curso básico de computación donde ha aprendidio a *chatear* y, como Carlos ha caído postrado en la cama por enfermedades asociadas a su avanzada edad, ha enseñado a sus amigos a chatear para "poder comunicarse sin tener que salir de casa". Pero esto no es más que una fachada para encubrir su engaño (y sin querer encubirir el de Beatriz también), pues Alicia quiere desea declararse *on-line* con Marcelo y de paso lo mismo hará Beatriz con Roberto.

Alicia, temerosa de ser descubierta y con ayuda del curso de computación, se ha dado cuenta de que cualquiera podría descubrir su infidelidad viendo cuales son los paquetes intercambiados entre el computador de Marcelo y ella. También ha estado pensando paranoicamente que Roberto podría estar tomando un curso de *hacker* en la municipalidad y podria intentar hacerse pasar por Marcelo e interferir en sus conversasiones privadas. Lo último que preocupa a Alicia es la influencia que pueden tener los otros programas que ejecutan el computador de ella o el de sus amigos.

Alicia ha determinado que su problema es exacatamente el siguiente:

Desea crear un protocolo en el que la comunicación es anonima y además es imoposible que alguien impersone a otro. Adicionalmente Alicia desea que las garantías anteriores se man-

tengan inclusive si el protocolo es ejecutado concurrentemente con otros protocolos
En esta memoria se desarrollará un protcolo criptográfico y se demostrará rigurosamente que cumple con las garantías necesaria para solucionar el problea anterior, utilizando herramientas modernas de Criptografía.

1.2. Descripción del problema

En general el problema anterior puede a aplicarse a cualquier de grupo de personas que desea comunicarse entre sí en una red (internet o una red local) y desea obtener garantías similares. A continuación iniciamos el camino de formalización del problema motivacional. La solución del problema consiste en encontrar un protocolo (un algoritmo distribuido) de cual se puedan garantizar matemáticamente las siguientes tres propiedades :

1. Anonimato
2. Autentificación desmentible
3. Componibilidad

Anonimato

A modo de ejemplo podemos considerar un protocolo "usual"de comunicación, un protcolo IP simplificado. En la figura 1.1 la figura cada flecha de A a B indica que A envió un mesaje a B. La etiqueta de una flecha de A a B indica el mensaje intercambiados en la ejecución de protocolo. Por ejemplo Roberto envió a Alicia el mensaje (m_{AR}, ip_R, ip_A), donde m_{AR} es el contenido del mensaje, ip_R es la dirección IP de Roberto y ip_A es la dirección IP de Alicia. Notemos que estos datos son necesarios para poder *rutear* los mensajes de un participante a otro, pero a la vez revelando a un adversario que Roberto envio un mensaje a Alicia. Por lo tanto podemos decir que **el protocolo IP simplificado no es anónimo** pues **existe un ataque**.

Para definir el anonimato resulta crucial definir formalmente que es considerado un ataque al anonimato, pues un protocolo será anónimo si y solo si no existe ningún ataque. En [3] se define un ataque con el siguiente juego. El adversario determinada dos posibles ejecuciones del protocolo, las cuales difieren en qué mensajes seran enviados por quién y qué mensajes

fueron recibidos por quién. Entonces consideraremos que el adversario realiza un ataque si al ejecutar al adversario con cada una de las posibles ejecuciones del protocolo, el adversario logra identificar cuál es. Por lo tanto un protocolo sera seguro si y solo si cualquier adversario no logra distinguir una ejecución de otra.

Autentificación desmentible

En el procolo de la figura 1.1 es posible que un adversario (rol que podria ser tomado por Marcelo) se haga pasar por Beatriz e intente comunicarse con Alicia, y para ello solo es necesario que modifique los mensajes uno de sus mensajes cambiando ip_M por ip_B. Por lo tanto decimos que el protocolo no implementa canales autenticados.

Con canales autentificado nos referimos protocolos en los cuales es posible estar seguro, con alta probabilidad, de quién es el autor de un mensaje. Sin embargo hay que ser cuidadoso con el protocolo de autentificación usado, pues los más conocidos (firmas digitales por ejemplo) poseen la propiedad de *non repudiability*. Esto es que el emisor de un mesaje autentificado no puede negar a "la comunidad"que el es el autor del mensaje. Esto estaría contradiciendo el anonimato, pues el adversario también sería capaz de asociar la autoria de un mensaje a el emisor de este.

La autentificacion desmentible, introducida en [2], se refiere a los protocolos que implementan canales anónimos con la propiedad adicional de que cada mensaje es autentificado a un receptor específico y el receptor no es capaz de probar a nadie más quién es el autor del mensaje.

Componibilidad

En general el hecho de implementar protocolos con ciertas garantías (por ejemplo anonimato y autentificación desmentible) no garantiza que dichas propiedades se sigan teniendo cuando el protocolo es ejecutado concurrentemente con otros protocolos.

En [1] Canetti introduce el *framework* criptografico conocido como *Universal Composability* (desde ahora UC). UC propone una metodología definir y demostrar los objetivos de seguridad de un protocolo (por ejemplo anonimato y autentificación) de un protocolo. UC garantiza que el protocol mantendrá su seguridad inclusive si es ejecutado concurrentemente con cualquier protocolo, siempre y cuando no comparta estado con el protocolo analizado.

Cuando el protocolo sí comparte estado con otros protocolo es necesario hacer uso del *frame-*

work Generalized Universal Composabillity (desde ahora GUC), que generaliza a UC. Informalmente GUC propone una metodología para incluir el estado que un protocolo podría compartir con otros.

Solución propuesta

En esta memoria proponemos un protocolo que permite solucinar nuestro problema motivacional utilizando técnicas de Anonimato y Autentificación desmentible y demostramos que el protocolo resuelve el problema utilizando el *framework* GUC. Es decir el protocolo soluciona el problema inclusive si es ejecutado concurrentemente con otros protocolos que pueden compartir estado con él.
En la figura 1.2 se muestra un diagrama para explicar nuestra solución.

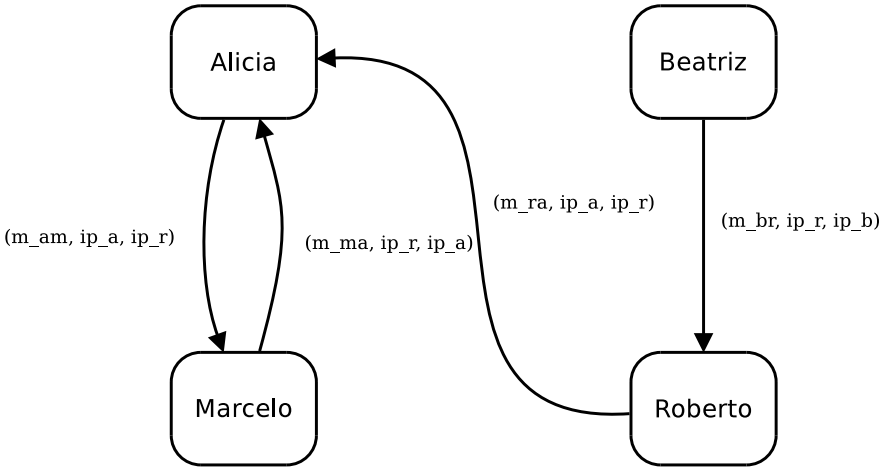

Figura 1.1: Protocolo simple de comunicación

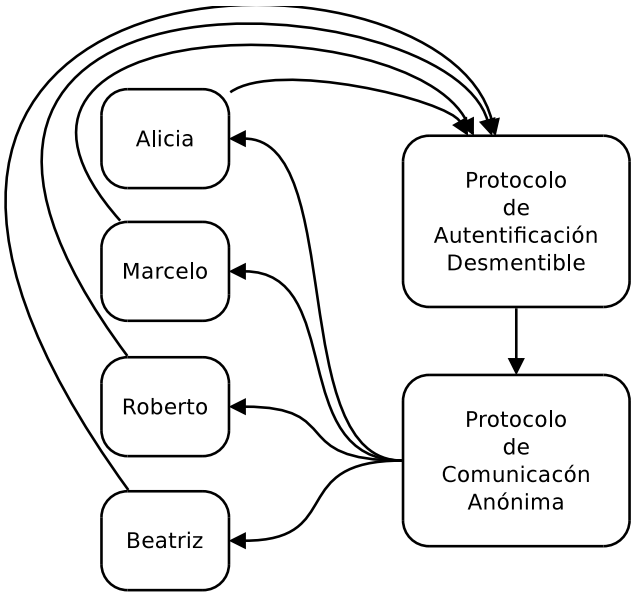

Figura 1.2: Solución propuesta

Capítulo 2

Marco Teórico

Capítulo 3

El protocolo

Capítulo 4

conclusiones

Referencias

[1] R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *FOCS*, pages 136–145, 2001.

[2] Dwork, Naor, and Sahai. Concurrent zero-knowledge. *JACM: Journal of the ACM*, 51, 2004.

[3] A. Hevia and D. Micciancio. An indistinguishability-based characterization of anonymous channels. In N. Borisov and I. Goldberg, editors, *Privacy Enhancing Technologies*, volume 5134 of *Lecture Notes in Computer Science*, pages 24–43. Springer, 2008.

Apéndices

A . Código clase `p2pSync`

B . Código clase `MessageManager`

C . Código clase `P2pPlayer`