UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MECANISMO PARA GARANTIA DE PRIVACIDADE EM REDES SOCIAIS ONLINE

RODRIGO PEREIRA BOTELHO

ORIENTADOR: PROF. DR. SERGIO DONIZETTI ZORZO

São Carlos - SP
Agosto/2011

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MECANISMO PARA GARANTIA DE PRIVACIDADE EM REDES SOCIAIS ONLINE

RODRIGO PEREIRA BOTELHO

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Sistemas Distribuídos e Redes
Orientador: Dr. Sergio Donizetti Zorzo

São Carlos - SP
Agosto/2011

Ficha catalográfica elaborada pelo DePT da Biblioteca Comunitária da UFSCar

B748mg

Botelho, Rodrigo Pereira.
Mecanismo para garantia de privacidade em redes sociais online / Rodrigo Pereira Botelho. -- São Carlos : UFSCar, 2012.
82 f.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2011.

1. Ciência da computação. 2. Redes sociais online. 3. Criptografia. 4. Privacidade e personalização. I. Título.

CDD: 004 (20ª)

Universidade Federal de São Carlos

Centro de Ciências Exatas e de Tecnologia

Programa de Pós-Graduação em Ciência da Computação

"Mecanismo para Garantia de Privacidade em Redes Sociais Online"

RODRIGO PEREIRA BOTELHO

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação

Membros da Banca:

Prof. Dr. Sergio Donizetti Zorzo
(Orientador - DC/UFSCar)

Prof. Dr. Hélio Crestana Guardia
(DC/UFSCar)

Profa. Dra. Cínthia Obladen de Almendra Freitas (PUC/PR)

São Carlos
Agosto/2011

Dedico este trabalho a Deus e a minha família.

AGRADECIMENTO

Agradeço aos professores do Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos pelo suporte, em especial ao meu orientador Sergio Donizetti Zorzo.

Agradeço aos meus amigos de trabalho Marcelo Filho, Marcelo Perecim, Mateus Pádua, Neemias Santos, Lucas Nascimento, Daniel Balieiro e Diego Volpe pelas discussões sobre Redes Socias/Privacidade e pelo esforço feito para realizarem as tarefas e cumprirem o cronograma do trabalho nas horas que eu estava dedicado ao mestrado.

Agradeço também ao Gustavo Hubaide, um exemplo de pessoa e de profissional com quem aprendi muito durante o tempo em que trabalhamos juntos. Gustavo, sem o seu apoio o caminho trilhado para este título seria mais árduo, muito obrigado.

O êxito na vida não se mede pelo que você conquistou, mas sim pelas dificuldades que superou no caminho.

Abraham Lincoln

RESUMO

A utilização de redes sociais online para o compartilhamento de dados está se tornando cada vez mais comum. Para se inscrever em uma rede social online, os usuários devem concordar com as políticas que estas redes impõem e isto implica, na maioria das vezes, em concordar que todos os dados postados podem ser utilizados para diversos fins, por exemplo, para a melhoria do serviço oferecido pela rede social. Ao mesmo tempo, as redes sociais online geralmente fornecem ferramentas para que o acesso aos dados compartilhados pelos usuários seja restrito. No entanto, esta restrição de acesso aplica-se apenas para usuários da rede social, possibilitando que terceiros e a própria rede social possam acessar e utilizar estes dados. Uma abordagem somente com criptografia dos dados é insuficiente para que a privacidade destes dados seja fornecida mantendo a possibilidade do usuário obter serviços personalizados. Este trabalho apresenta um mecanismo baseado em um formato de mensagens bem conhecido juntamente com a utilização de métodos de criptografia simétrica e assimétrica para estender a privacidade em redes sociais online, garantindo a privacidade de certos tipos de dados dos usuários em relação a outros usuários da rede, a terceiros e a própria rede social, possibilitando mesmo assim a aplicação de serviços personalizados para os usuários da rede social. Adicionalmente, um estudo de caso com a utilização do mecanismo em uma rede social online real e uma análise sobre o mecanismo são apresentados.

Palavras-chave: Redes Sociais Online, Privacidade, Criptografia, Personalização.

ABSTRACT

The use of online social networks for sharing personal data is becoming increasingly common. To join an online social network, users must agree to the online social networks policies and this generally implies agreeing that all posted data can be used for various purposes, for instance, to improve the service offered by the social network operator. Online social networks generally provide tools to allow users to restrict access to their shared data. However, these access restrictions applies only to social network users, enabling third parties and the social network itself access and use to these data. A data encryption based approach only is insufficient to ensure privacy and preserve the ability of the user to obtain personalized services. This paper presents a mechanism based on a well-known message format coupled with the use of methods for symmetric and asymmetric encryption to extend the privacy in online social networks, ensuring that certain user data types remain private to other network users, third parties and own social network, yet allowing the offering of personalized services. Additionally, a case study using the mechanism in a real online social network and an analysis of the mechanism are presented.

Keywords: Online Social Networks, Privacy, Encryption, Personalization.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AES - *Advanced Encryption Standard*

API - *Application Programming Interface*

ABE - *Attribute-Based Encryption*

DOM - *Document Object Model*

XML - *Extensible Markup Language*

HTML - *HyperText Markup Language*

HTTPS - *HyperText Transfer Protocol Secure*

JSON - *JavaScript Object Notation*

RAM - *Random Access Memory*

REST - *Representational State Transfer*

RSA - *Rivest, Shamir e Adleman*

URL - *Uniform Resource Locator*

XPath - *XML Path*

SUMÁRIO

Capítulo 1

INTRODUÇÃO

Este capítulo apresenta uma introdução a redes sociais online e aos riscos envolvidos no compartilhamento de informações pessoais nestes ambientes. Também são apresentados as motivações, objetivos e organização do trabalho.

1.1 Contexto

As redes sociais online, como o Facebook[1] e o Orkut[2], são caracterizadas por possibilitar que os usuários criem um perfil público, mantenha uma lista de amigos possibilitando a navegação por esta lista e pela interação com os amigos (BOYD; ELLISON, 2008). Estas redes sociais online são utilizadas por milhões de usuários no mundo (AHN *et al.*, 2007; MISLOVE *et al.*, 2007). Os motivos que levam os usuários a participarem destas redes sociais são diversos, por exemplo, manter contato com colegas de classe, organizar movimentos em prol de algum objetivo comum, assuntos técnicos e profissionais, entre outros (BOYD; ELLISON, 2008; ELLISON *et al.*, 2009).

Para que estas interações sociais sejam possíveis, os usuários normalmente disponibilizam uma série de informações pessoais, que podem incluir nome, endereços de e-mail, visões políticas, orientação sexual, entre outras. A revelação de certas informações pode ser controlada, pois as redes sociais online geralmente disponibilizam ferramentas para que os usuários configurem a visibilidade dos

[1] Disponível em: <http://www.facebook.com/>. Acesso em: 13/06/2011.

[2] Disponível em: <http://www.orkut.com.br/>. Acesso em: 13/06/2011.

dados. Porém, os usuários despreocupados com a privacidade dos dados pessoais tendem a deixar as configurações iniciais, que geralmente concedem uma visibilidade total dos dados para os usuários da rede social online (TOCH;SADEH *et al.*, 2010). Outro ponto importante é que geralmente as redes sociais não oferecem mecanismos para controlar o acesso aos dados em relação a terceiros e em relação à própria rede social.

Além disto, é comum encontrar nas políticas de uso das redes sociais online, termos que garantem o uso lícito dos dados pessoais dos usuários para melhoria de serviço. Uso lícito neste caso está relacionado com a forma que os dados do usuário são utilizados, fornecendo garantias que os dados dos usuários não serão utilizados com propósitos diferentes dos quais foram acordados previamente. Considerando que as redes sociais online geralmente oferecem o serviço gratuitamente ao usuário, dados agregados dos usuários podem ser utilizados para fins de propaganda direcionada. Desta forma, os anunciantes e não os usuários tornam a manutenção da rede social viável, no que diz respeito às questões financeiras.

Adicionalmente, dados pessoais dos usuários podem ser revelados para aplicações sociais desenvolvidas por terceiros. As aplicações sociais são geralmente jogos e utilitários que podem ser instalados pelo usuário. Como dito anteriormente, as redes sociais online não oferecem ferramentas para o usuário configurar o acesso aos dados em relação a terceiros. Portanto, os usuários que instalam as aplicações sociais estão consentindo que os dados pessoais sejam compartilhados com tais aplicações.

Neste contexto, a revelação dos dados pessoais dos usuários é tratada como um problema e apresentada como motivação do trabalho na seção 1.2.

1.2 Motivação e Objetivos

A revelação de dados pessoais pode trazer diversos riscos para o usuário tanto de maneira *online* quanto de maneira *offline.* Por exemplo, um usuário pode ser vítima de *phishing*, que por meio da utilização de dados disponibilizados em redes sociais pode se tornar mais efetivo; roubo de bens materiais – pela combinação dos dados um criminoso pode determinar a rotina do usuário para uma

ação, entre outros. Portanto, é importante que os usuários tomem conhecimento destes riscos. Neste sentido ferramentas que possam prevenir tais ações, a utilização dos dados do usuário com propósitos diferentes dos que o usuário consentiu, são essenciais.

Assim, o objetivo geral deste trabalho é propor um mecanismo baseado em mensagens criptografadas para que a privacidade dos usuários seja garantida em casos de invasão e roubo de dados do servidor da rede social e *crawling* dos dados da rede social. Para isto a privacidade é garantida em alguns aspectos, por exemplo, dados pessoais do perfil e conteúdo textual das interações entre usuários. Além disto, o mecanismo deve manter a utilidade dos dados pessoais do usuário, possibilitando que serviços personalizados sejam oferecidos ao usuário.

Para que este objetivo geral seja alcançado, os objetivos específicos a seguir podem ser citados: (a) mínimo de alterações para que a rede social e terceiros possam implementar o mecanismo, (b) não implicar em reduções de serviço e não afetar serviços que possam ser oferecidos futuramente e (c) que a interferência na usabilidade da rede social seja mínima.

1.3 Organização do Trabalho

Esta seção apresenta a organização do trabalho, destacando uma visão geral sobre cada capítulo.

No Capítulo 2 são apresentados alguns assuntos relacionados a redes sociais online e privacidade. O capítulo faz uma introdução sobre redes sociais online e os tipos de redes sociais online existentes, destacando elementos que caracterizam as redes sociais online e os serviços oferecidos. A privacidade na visão do usuário e os riscos de violação de privacidade envolvidos no compartilhamento de informações pessoais também são abordados no Capítulo 2, bem como algumas ferramentas de privacidade.

O Capítulo 3 apresenta o mecanismo de privacidade proposto neste trabalho. São levantados os requisitos funcionais e apresentado o mecanismo proposto – um mecanismo baseado em mensagens cifradas. Adicionalmente, uma implementação do mecanismo é apresentada.

O Capítulo 4 apresenta uma análise do mecanismo, introduzindo um estudo de caso que leva em consideração um usuário protegido pelo mecanismo apresentado no Capítulo 3. Uma análise de sobrecarga de tempo e armazenamento também é feita no Capítulo 4, assim como uma análise do mecanismo e suas limitações.

No Capítulo 5 são apresentados os trabalhos relacionados que tem como objetivo proteger a privacidade total ou parcial dos usuários em redes sociais online.

Por fim, o Capítulo 6 apresenta a conclusão do trabalho com as contribuições e sugestões de trabalhos futuros.

Capítulo 2

REDES SOCIAIS ONLINE E PRIVACIDADE

Este capítulo apresenta uma introdução sobre as redes sociais online e aos riscos de violação de privacidade no compartilhamento de dados pessoais nestas redes.

2.1 Considerações Iniciais

Neste capítulo são apresentados alguns aspectos sobre as redes sociais como a sua estrutura e características. É apresentada a noção de como o usuário se inscreve e compartilha os dados pessoais em redes sociais online de diversos tipos, bem como assuntos relacionados à privacidade dos usuários como, por exemplo, a liberação de dados do usuário para aplicações sociais. Adicionalmente, são levantados alguns aspectos que podem implicar na violação de privacidade dos usuários e algumas ferramentas de privacidade que permitem melhorar a privacidade em relação a estes riscos.

O capítulo está organizado da seguinte forma: a seção 2.2 apresenta uma introdução sobre as redes sociais online, na seção 2.3 são apresentados alguns tipos de redes sociais online, a seção 2.4 faz uma introdução sobre as aplicações sociais, a privacidade no ponto de vista do usuário é apresentada na seção 2.5, na seção 2.6 são apresentados alguns riscos de violação de privacidade no compartilhamento de dados pessoais, a seção 2.7 apresenta algumas ferramentas e abordagens que permitem manter a privacidade do usuário em redes sociais online e a seção 2.8 encerra o capítulo com as considerações finais.

2.2 Redes Sociais Online

As redes sociais online atraem em seus sites milhões de usuários diariamente, que as utilizam para os mais diversos fins, por exemplo, marcar encontros, manter contatos profissionais, compartilhar lugares que estão, entre outros. Assim, os usuários utilizam as redes sociais principalmente para a comunicação e compartilhamento de informações com amigos. O termo “amigos” é utilizado para representar um contato do usuário na rede social, porém o contato pode ter outro tipo de relação com o usuário na vida real, por exemplo, irmão, mãe, primo, entre outros (BOYD, 2006).

Alguns tipos de redes sociais permitem que os usuários conectem-se sem aprovação mútua e outros tipos de redes sociais requerem a aprovação de ambos os usuários para que estes tenham a ligação de amigos. Além da ligação de amigo, algumas redes sociais possibilitam o que é chamado de “seguidores”. Trata-se de um tipo de ligação não simétrica, que possibilita um usuário acessar os conteúdos compartilhados por outro sem a necessidade de haver reciprocidade.

Um serviço pode ser considerado uma rede social online caso permita que os usuários criem um perfil público possibilitando que os participantes da rede social possam visualizá-lo, permita que o usuário mantenha uma lista de amigos possibilitando a navegação por esta lista e permita a interação com estes amigos, por exemplo, troca de mensagens (BOYD; ELLISON, 2008). No perfil, um usuário pode disponibilizar fotos, informações pessoais, visões políticas, informações profissionais, entre outras.

Além do perfil, listas de amigos e a possibilidade de interação com amigos, algumas redes sociais disponibilizam *Application Programming Interface* (API) para que terceiros possam desenvolver aplicações sociais que possibilitam estender a funcionalidade das redes. Estas aplicações sociais são, em sua grande maioria, utilitários e jogos, que possuem acesso aos dados do perfil do usuário e à lista de amigos, e oferecem em troca o entreterimento.

Outro tipo de serviço bastante comum oferecido pelas redes sociais é o de publicidade direcionada. Dado que as redes sociais possuem bastante informação sobre seus usuários, a prática desta atividade pode ser bastante efetiva, ajudando um vendedor a encontrar um cliente e vice-versa.

Uma preocupação inerente diante da facilidade de compartilhamento de informações sobre um usuário é a privacidade destes dados. Para isto, as redes sociais protegem a privacidade dos usuários por meio de suas políticas de privacidade. No entanto, mesmo confiando nas políticas de privacidade os usuários estão sujeitos a terem seus dados compartilhados sendo utilizados de forma indevida, como é discutido na seção 2.6.

Outro ponto importante é que ao aceitar as políticas das redes sociais, o usuário concede o direito sobre os dados compartilhados, permitindo que as redes sociais atuem como autoras do conteúdo. Uma alternativa para isto é a possibilidade descentralizar os dados, obtendo uma rede social descentralizada. Uma combinação de computação em nuvem e de computadores pessoais servindo conteúdo pode ser capaz de substituir as redes sociais centralizadas garantindo a autoria e privacidade dos dados para os usuários (SHAKIMOV *et al.*, 2009). Porém, ainda hoje as redes sociais centralizadas continuam sendo as mais populares como, por exemplo, o Facebook e o Orkut.

Entre os dados compartilhados pelos usuários geralmente estão fotos, vídeos e dados sensitivos[3] como, por exemplo, endereço de e-mail, telefone, entre outros. Claramente estes tipos de dados estão relacionados com a natureza da rede social. A seção 2.3 identifica alguns tipos de redes sociais online e apresenta suas características.

2.3 Tipos de Redes Sociais Online

É comum notar que alguns usuários reconheçam somente as redes sociais de propósito gerais, como o Orkut e Facebook, como redes sociais. Porém, estas são apenas um tipo de rede social e esta seção identifica outros, apresentando as principais características de cada tipo de rede social.

Como dito na seção anterior, uma rede social é caracterizada pela possibilidade de criar um perfil de usuário, manter a lista de amigos ou seguidores e permitir a interação entre usuários. Desta forma, alguns serviços de

[3] Entende-se como dados sensitivos, no contexto deste trabalho, todos os dados que podem levar a identificação do usuário.

compartilhamento de conteúdo também podem ser considerados como rede social e tipificados de acordo com o serviço que disponibilizam. Entre estes serviços pode-se citar os tipos de compartilhamento de vídeo, fotos, geolocalização, informações profissionais, informações de saúde, entre outros.

Os serviços de compartilhamento de vídeo permitem que um usuário além de compartilhar vídeos, possa efetuar anotações, buscar, comentar, adicionar palavras chave para busca e criar relacionamento entre vídeos. Entre os serviços de compartilhamento de vídeos com características de redes sociais pode-se citar, entre outros, Youtube[4] e Vimeo[5].

De maneira parecida, serviços de compartilhamento de fotos também possibilitam comentar, anotar e buscar fotos. Alguns serviços de compartilhamento de fotos também possuem integração com outras redes sociais, o que permite compartilhar fotos que estão armazenadas pelo serviço diretamente na rede social, como é o caso do Flickr[6] e DeviantArt[7].

Os serviços de geolocalização geralmente fornecem um aplicativo que pode ser instalado em dispositivos móveis. O objetivo é informar a localização do usuário para seus amigos, por exemplo, chegar ao colégio, em um local público, entre outros. Exemplos de serviços de geolocalização que podem ser considerados como rede social são Foursquare[8] e Locaccino (TOCH; CRANSHAW *et al.*, 2010).

As redes sociais para contatos profissionais, como o LinkedIn[9], permitem que o usuário mantenha contatos profissionais de empresas onde trabalha e de onde trabalhou. Além disto, permite a troca de informações profissionais como, por exemplo, métodos para a resolução de um determinado problema, tecnologias utilizadas, entre outros.

As redes sociais de saúde permitem que usuários compartilhem o histórico de saúde com entidades de saúde, por exemplo, um médico ou um hospital (WILLIAMS, 2010). Entre as informações compartilhadas podem estar resultados de

[4] Disponível em: <http://www.youtube.com/>. Acesso em: 13/06/2011.

[5] Disponível em: <http://www.vimeo.com/>. Acesso em: 13/06/2011.

[6] Disponível em: <http://www.flickr.com/>. Acesso em: 13/06/2011.

[7] Diponível em: <http://www.deviantart.com/>. Acesso em: 13/06/2011.

[8] Disponível em: <https://foursquare.com/>. Acesso em: 13/06/2011.

[9] Disponível em: <http://www.linkedin.com/>. Acesso em: 13/06/2011.

exames, contra-indicações de remédios, alergias, entre outros. Exemplos de redes sociais deste tipo incluem Google Health[10] e Microsoft HealthVault[11].

Por fim, pode-se citar alguns serviços bem conhecidos que também podem ser considerados como tipo de redes sociais. É o caso de fóruns de discussão e blogs. Os fóruns de discussão permitem que os usuários conversem sobre tópicos específicos e compartilhem suas experiências sobre um determinado assunto. Os blogs permitem que usuários sejam autores de diversos tipos de conteúdos, por exemplo, tecnologia, política, entre outros. De maneira parecida com os blogs, os microblogs se diferenciam por tornar os conteúdos mais objetivos, geralmente referenciando outros conteúdos como, por exemplo, o Twitter[12].

2.4 Aplicações Sociais

Os usuários de redes sociais podem disponibilizar no perfil informações pessoais, fotos, visões políticas, algumas informações profissionais, entre outras. Além disto, podem instalar no perfil, aplicações sociais que são programas de terceiros que fornecem algum tipo de serviço personalizado.

Alguns tipos comuns de aplicativos para redes sociais são jogos e utilitários. Os jogos geralmente permitem ao usuário interagir com seus amigos da rede social e evoluir seu *avatar*[13]. Por outro lado, os utilitários geralmente apresentam algum tipo de informação personalizada para o usuário como, por exemplo, horóscopo.

As aplicações sociais geralmente são oferecidas em redes sociais de propósito geral, mas não há restrições para o oferecimento em redes sociais de outro tipo. Para que terceiros possam desenvolver as aplicações, as redes sociais fornecem APIs que possibilitam o acesso a certas informações do usuário, por exemplo, foto do perfil, algumas informações pessoais do perfil, lista de amigos, e algumas informações do perfil dos amigos.

[10] Disponível em: <http://www.google.com/health/>. Acesso em 13/06/2011.

[11] Disponível em: <http://www.microsoft.com/en-us/healthvault/>. Acesso em 13/06/2011.

[12] Disponível em: <http://twitter.com/>. Acesso em: 13/06/2011.

[13] Avatar é o termo utilizado para representar o personagem do usuário no jogo.

Dada a quantidade de dados que as aplicações sociais podem acessar de maneira lícita, as redes sociais restringem o uso destes dados por meio das políticas para desenvolvimento destas aplicações. As políticas variam de site para site, mas geralmente restringem o uso abusivo dos dados, como o armazenamento permanente e venda dos dados, bem como a utilização para outros fins que são diferentes do propósito inicial descrito pelos desenvolvedores. Porém, desenvolvedores podem ainda assim desrespeitar estas políticas e utilizar os dados de maneira ilícita. Alguns dos riscos de violação de privacidade que os usuários estão submetidos são apresentados na seção 2.6.

Para instalar uma aplicação social no perfil, os usuários também devem concordar com uma política de privacidade que informa revelar para a aplicação que está sendo instalada, dados do usuário ativo além de alguns dados dos amigos. Isto significa que, uma aplicação social pode ter acesso a alguns dados do perfil do usuário, como o nome e foto do perfil, mesmo que o usuário em questão não tenha instalado a aplicação, bastando apenas que um de seus amigos tenha feito.

A Figura 2.1 apresenta a arquitetura geralmente utilizada pelas redes sociais para que terceiros desenvolvam aplicações sociais.

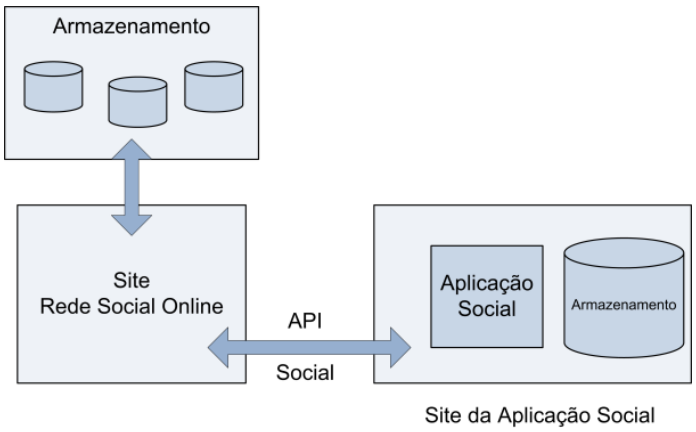


Figura 2.1 – Arquitetura para Desenvolvimento de Aplicações Sociais.

A aplicação social é hospedada no servidor *Web* do desenvolvedor, lado direito da Figura 2.1. Este servidor pode ter um serviço de armazenamento próprio, que é utilizado caso o desenvolvedor opte por armazenar alguns dados do usuário em *cache*. Isto evita a utilização da API Social para obter dados do usuário e em alguns casos esta abordagem pode diminuir o tempo de recuperação dos dados.

Caso os dados não estejam em *cache*, podem ser requisitados pela aplicação social por meio da API social. Então, os dados são recuperados pela rede social e fornecidos para a aplicação social.

A abordagem com *cache* de dados não é obrigatória, mas se for utilizada, o *cache* deve ser mantido apenas por um período tempo que é determinado pela rede social e após este tempo deve ser excluído permanentemente. Embora esta seja uma restrição geralmente aplicada nas políticas de desenvolvimento de aplicações sociais, pode-se perceber que as aplicações podem continuar armazenando indevidamente os dados dos usuários. A seção 2.5 apresenta como os usuários percebem estas possíveis revelações de privacidade.

2.5 Privacidade em Redes Sociais

A privacidade é um assunto importante, pois possibilita que os usuários controlem a exposição e compartilhamento de seus dados pessoais. Ao utilizar qualquer serviço online o usuário tem direito de manter sua privacidade, ter ciência de como e onde seus dados pessoais estão sendo utilizados. Porém, em casos de crimes digitais, os serviços online devem ser capazes de identificar e responsabilizar os infratores. Este é um desafio comum em serviços online, oferecer personalização mantendo a privacidade do usuário, mas com a possibilidade de identificar infratores em caso de crimes digitais para que as autoridades possam tomar as devidas providencias perante a lei vigente no local do crime.

Esta seção discute alguns pontos em relação à percepção de privacidade que alguns usuários podem ter no que diz respeito à liberação de dados pessoais para as redes sociais.

Alguns usuários acreditam que as informações pessoais fornecidas para as redes sociais não os prejudicarão no futuro e caso possam perceber que isto os afetaria de alguma forma, poderiam apagar tais informações do perfil. Outro pensamento comum é que se alguns dados são compartilhados, então não há problema caso terceiros os utilizem (CONTI; SOBIESK, 2007; SOLOVE, 2007).

No entanto, as visões dos usuários a respeito de um determinado assunto podem mudar no futuro e os dados compartilhados podem ter sido disseminados,

tornando possível o acesso a estes dados, mesmo que o usuário apague o conteúdo original. Levando em consideração que está se tornando comum empresas utilizarem as redes sociais para avaliar perfil dos candidatos no processo de seleção, o compartilhamento de certos dados pessoais pode influenciar, por exemplo, na escolha de um funcionário em relação a outro.

Restringir o acesso aos dados seria uma alternativa para diminuir os riscos, porém alguns usuários permitem que os dados pessoais sejam vistos por todos simplesmente por não terem idéia de quem pode acessá-los (KRISHNAMURTHY; WILLS, 2008). Assim deixam as preferências de privacidade oferecidas pelas redes sociais no nível padrão (GROSS; ACQUISTI, 2005), que geralmente permite um número maior de usuários com direito de acesso ao conteúdo compartilhado.

Por outro lado, não são todos os usuários que correm certos riscos de violação de privacidade por compartilharem informações pessoais. Os usuários que têm maior confiança na rede social (DWYER *et al.*, 2007) e o maior número de amigos na rede (YOUNG; QUAN-HAASE, 2009) tendem a compartilhar mais informações. Em paralelo a isto, usuários com maior experiência e ciência sobre os riscos de violação de privacidade na Internet, tendem utilizar as preferências de privacidade para restringir o acesso a determinados tipos de dados (YOUNG; QUAN-HAASE, 2009). Além disto, de acordo com Schrammel *et al.* (2009), informações demográficas dos usuários como idade e sexo não são variáveis relevantes para justificar o compartilhamento de dados pessoais, porém os autores notaram que usuários que estão no mercado de trabalho tendem a revelar menos informações.

Os motivos que levam os usuários a compartilhar as informações pessoais nas redes sociais são diversos, por exemplo, manter informado um amigo distante sobre algumas atividades que o usuário está desempenhando, publicar a situação do relacionamento para começar ou evitar um novo namoro, entre outros. No entanto, Young e Quan-Haase (2009) perceberam que usuários já estão restringindo acesso a alguns tipos de informações como, por exemplo, endereço de e-mail e telefone.

Desta forma, pode-se perceber que usuários estão começando a se preocupar mais com a privacidade em redes sociais online e a tendência é que os usuários façam isso à medida que tomem conhecimento dos riscos envolvidos com a revelação de informações pessoais, conforme é discutido na próxima seção.

2.6 Riscos no Compartilhamento de Dados Pessoais

Na seção 2.5 foi citada uma possível utilização dos dados pessoais compartilhados por um usuário em um processo de seleção para um emprego. Embora este exemplo possa não apresentar um grande risco, dependendo da natureza dos dados compartilhados o usuário pode ser exposto a crimes físicos e cibernéticos (GROSS; ACQUISTI, 2005) como, por exemplo, roubo de bens materiais, roubo de identidade, roubo de informações bancárias, entre outros. Isto se torna possível, pois em alguns casos são reveladas informações que somente contatos próximos saberiam (GEORGE, 2006), permitindo que os ataques possam ser mais efetivos.

Para desempenhar estes ataques com efetividade, diversas técnicas são utilizadas para obter e combinar a maior quantidade de dados possível sobre um usuário, por exemplo, ataques de intersecção, de identificação e de inferência (CHEN; SHI, 2009). Os ataques de intersecção combinam conjuntos de dados à procura por atributos com valores exclusivos que permitem efetuar a identificação de um usuário. Ataques de identificação são feitos para determinar identidades dos usuários na vida real a partir de dados contidos nas redes sociais. Por fim, os ataques de inferência utilizam técnicas de mineração de dados para descobrir dados e padrões escondidos em redes sociais.

A partir das técnicas citadas anteriormente, pode-se obter um perfil do usuário mais completo e estes dados pessoais dos usuários podem ser utilizados para diversos fins como, por exemplo, *spam* (BROWN *et al.*, 2008; HUBER; MULAZZANI; SCHRITTWIESER *et al.*, 2010; HUBER; MULAZZANI; WEIPPL *et al.*, 2010; KAAFAR; MANILS, 2010; POLAKIS *et al.*, 2010). Este tipo de prática, *spam* social, possui algumas propriedades interessantes, como as preferências do usuário em relação a determinado assunto, que podem ser combinadas com algumas informações sociais para atacar vítimas (JAGATIC *et al.*, 2007). Por exemplo, um e-mail de um falso amigo íntimo pode levar um usuário legítimo revelar informações para terceiros.

Para que os ataques sociais citados anteriormente possam acontecer, pode-se notar que geralmente é feita uma combinação entre as informações pessoais e as relações sociais do usuário. Como geralmente a exibição e a possibilidade de

navegação pela lista de amigos são recursos disponíveis para todos os usuários de uma rede social, um adversário[14] pode obter a lista de amigos de um determinado usuário de maneira facilitada. No entanto, Korolova *et al.* (2008) argumentam que o ideal seria esconder a lista de amigos ou então mostrar um número limitado para que isto não pudesse comprometer a privacidade dos usuários. Porém, é possível quebrar a privacidade de usuários mesmo em casos nos quais o acesso à lista de amigos seja restrito (ASUNCION; GOODRICH, 2010), bem como inferir dados privados do perfil do usuário, baseando-se em dados de lista de amigos e grupos que o usuário participa (ZHELEVA; GETOOR, 2009).

Embora o círculo social *online* possa ser diferente do círculo social na vida real (WILSON *et al.*, 2009), pode-se utilizar técnicas para identificar amigos da rede social online que um usuário tem uma relação mais próxima. Por exemplo, algumas redes sociais permitem que usuários enviem e organizem fotos, além de possibilitar que usuários relacionem os amigos presentes na foto. Desta forma, o recurso de anotações em fotos permite que um adversário identifique quais usuários do círculo social *online* também fazem parte do círculo social na vida real (BESMER; LIPFORD, 2009; BESMER; LIPFORD, 2010), por meio de uma verificação de quais usuários estão marcados em uma foto.

O mecanismo proposto neste trabalho elimina alguns dos riscos apresentados nesta seção, pois garante a privacidade dos usuários contra ataque de invasão do servidor da rede social e de *crawling* dos dados do usuário. Outras ferramentas podem garantir a privacidade em outros aspectos, como é apresentado na seção 2.7.

2.7 Ferramentas de Privacidade

Como discutido na seção 2.6, os usuários correm diversos riscos ao publicar informações pessoais nas redes sociais. Por exemplo, estão mais vulneráveis a ataques de *phishing*, pois os adversários podem contatar os usuários em nome de um amigo íntimo. Também é possível roubo de bens materiais, pois os adversários

[14] Adversário, neste contexto, refere-se ao usuário que tem a intenção de obter os dados dos usuários para utilização ilícita.

podem saber a localização onde o usuário está no momento, entre outros fatos. Adicionalmente, foi discutido que, para isto, os adversários podem reunir e combinar dados dos usuários de diversas fontes para aumentar a efetividade dos ataques.

Os dados dos usuários que estão nas redes sociais geralmente podem ser obtidos pelos adversários de três formas. O primeiro cenário é uma rede social que liberou parte ou a totalidade do grafo social[15], aplicando técnicas para tornar o grafo anônimo. Os outros dois cenários implicam que os adversários obtiveram os dados por meio de técnicas de *crawling* ou invasão de servidores da rede social. Esta seção apresenta as ferramentas para garantir a privacidade em relação aos ataques contra o grafo social e *crawling*, enquanto o Capítulo 3 e o Capítulo 5 apresentam abordagens para proteger contra *crawling* e invasão.

Puttaswamy *et al.* (2009) utilizam técnicas para modificar o grafo social original, mantendo k usuários idênticos em um determinado sub-grafo, o que protege contra ataques de intersecção. De maneira parecida, Hay *et al.* (2007) também propõem uma extensão do grafo social original, modificando a estrutura de arestas para proteger os usuários contra ataques de identificação. Para os ataques de inferência, Zheleva e Getoor (2009) e Lindamood *et al.* (2009) propõem técnicas para remoção de informações e de ligações entre vértices dos grafos sociais que possam levar à revelação da privacidade do usuário.

Para proteger os dados do usuário contra a obtenção por meio de *crawling*, uma configuração correta nas preferências de privacidade seria o suficiente. Porém, o formato das políticas de privacidade e a forma de apresentação geralmente não são facilmente compreendidos pelos usuários (POLLACH, 2007) e algumas vezes isto pode levar o usuário manter as preferências de privacidade no nível padrão (TOCH;SADEH *et al.*, 2010), que geralmente torna pública grande parte dos dados do usuário.

Desta forma, as preferências de privacidade deveriam ser personalizadas de acordo com o usuário e não como um formato padrão para todos. Abordagens que extraem informações automaticamente do perfil do usuário (DANEZIS, 2009; TOCH;SADEH *et al.*, 2010) ou que obtêm manualmente alguns parâmetros de

[15] Trata-se de um grafo que representa as relações sociais de um usuário. Por exemplo, a representação da conexão de amizade entre os usuários de uma rede social pode ser feita por um grafo social no qual os vértices representam os usuários e as arestas representam a existência de amizade entre os dois usuários.

entrada (FANG; LEFEVRE, 2010) para sugerir preferências de privacidade mostram-se como alternativas. Por fim, embora seja uma tarefa não trivial, usuários poderiam contribuir verificando se as políticas de privacidade estão sendo aplicadas de maneira adequada (PETTERSSON *et al.*, 2006) e tomar providências caso não estejam.

2.8 Considerações Finais

Este capítulo apresentou uma introdução às redes sociais online, à estrutura e ao funcionamento geral destas redes. Foram levantados alguns aspectos de privacidade em relação aos usuários que utilizam as redes sociais online, bem como os riscos e ferramentas de privacidade. O próximo capítulo apresenta um mecanismo para estender a privacidade dos usuários em redes sociais *online*, possibilitando que serviços personalizados continuem sendo oferecidos aos usuários.

Capítulo 3

MECANISMO PARA GARANTIA DE PRIVACIDADE EM REDES SOCIAIS ONLINE

O mecanismo apresentado neste capítulo baseia-se em métodos de criptografia para estender a privacidade em redes sociais online. É apresentada também a implementação prova de conceito do mecanismo.

3.1 Considerações Iniciais

Este capítulo apresenta as partes que compõem o mecanismo desenvolvido, que tem como objetivo estender a privacidade das redes sociais online, prevenindo que dados pessoais – por exemplo, nome, telefone, interações textuais com amigos, entre outros – sejam utilizados por terceiros sem o consentimento do usuário. Cada componente é detalhado, sendo apresentadas as responsabilidades e as interações efetuadas. Os requisitos que o mecanismo deve contemplar para interferir o mínimo possível na utilização da rede social pelos usuários também são levantados, bem como uma aplicação prova de conceito que implementa o mecanismo.

O capítulo está organizado da seguinte forma: a seção 3.2 apresenta os requisitos que o mecanismo deve atender, na seção 3.3 o mecanismo é apresentado, a seção 3.4 apresenta a implementação do mecanismo e o capítulo é encerrado na seção 3.5 com as considerações finais.

3.2 Requisitos

Esta seção apresenta alguns requisitos da proposta, que podem justificar uma decisão tomada na implementação do mecanismo ou que podem servir para tornar o mecanismo menos intrusivo no que diz respeito à usabilidade da rede social por parte do usuário.

O primeiro ponto importante a ser levado em consideração é a maneira em que as redes sociais online são acessadas hoje. O acesso pode ser feito a partir de qualquer dispositivo que tenha acesso à Internet e consiga interpretar páginas *Web*, considerando que a maioria ou todas as redes sociais são aplicações *Web*. Portanto, o mecanismo não deve impor restrições de acesso neste sentido.

Outro ponto é que a comunicação dos usuários das redes sociais por meio do *Privacify*, o mecanismo proposto neste trabalho, só deve ser possível entre usuários que já estão utilizando a ferramenta. Isto significa que um usuário não deve ser capaz de enviar mensagens protegidas do *Privacify* para outros usuários que não estejam utilizando o *Privacify*. Com esta restrição, evita-se que um usuário seja incapaz de ler uma mensagem e mantém-se a usabilidade normal da rede social.

O usuário também deve possuir a opção de não utilizar mensagens protegidas em determinadas situações, mesmo que seja para uma comunicação entre dois usuários que estejam utilizando o *Privacify*. Embora isto não seja recomendável, é necessário, pois pode ocorrer a situação em que alguns destinatários da mensagem simplesmente não queiram utilizar o *Privacify*. Deixando o usuário livre para escolher se quer enviar uma mensagem protegida, não se exige que todos os amigos do usuário na rede social instalem a ferramenta e adotem o mecanismo para que possa ver um determinado conteúdo, mantendo o fluxo de comunicação normal e sem a proteção adicional fornecida pelo *Privacify*.

Além disto, deve haver a possibilidade de o usuário efetuar a configuração do nível de proteção dos dados, permitindo um balanço entre a qualidade do serviço oferecido e a privacidade garantida.

Por fim, o mecanismo deve permitir que os dados baseados em texto dos usuários continuem sendo utilizados pelas redes sociais e por terceiros para fins de personalização e melhoria do serviço. Entre as possibilidades de utilização dos

dados por terceiros pode-se incluir propagandas direcionadas de produtos, aplicações sociais, entre outros.

Os requisitos citados anteriormente nesta seção são levados em consideração e contemplados no mecanismo apresentado na seção 3.3.

3.3 Mecanismo de Privacidade Proposto

Esta seção apresenta o mecanismo proposto conforme os requisitos apresentados na seção 3.2, detalhando os componentes do mecanismo e listando as responsabilidades e interações de cada item.

A Figura 3.1 apresenta os componentes do mecanismo, ilustrando em alto nível a comunicação entre eles. Do lado esquerdo da Figura 3.1 estão representados os *Usuários*, que podem acessar a rede social por meio de *Clientes Privacify*. Desta forma, o papel dos *Clientes Privacify* é fornecer acesso à rede social e implementar o mecanismo proposto. Na prática, caso a rede social seja uma aplicação *Web*, o componente *Cliente Privacify* pode ser um navegador *Web* com um complemento que implemente o mecanismo do *Privacify*.

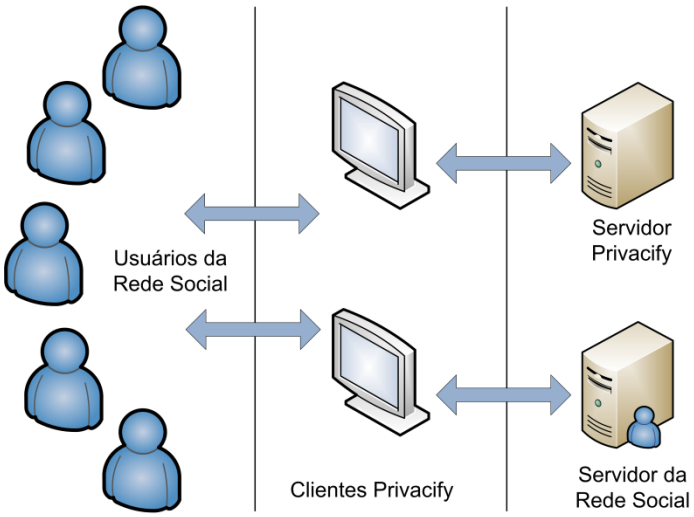


Figura 3.1 – Componentes do Mecanismo.

Para que os *Clientes Privacify* possam cumprir seu papel é necessária a comunicação – por um canal seguro – com o *Servidor da Rede Social* e com o

Servidor Privacify, ilustrado na parte direita da Figura 3.1. A comunicação com o *Servidor Privacify* é necessária para a recuperação de certas informações que possibilitam a leitura correta, por parte dos usuários, de mensagens protegidas enviadas para a rede social. Pode-se observar também na Figura 3.1, que o *Servidor da Rede Social* não se comunica diretamente com o *Servidor Privacify*, logo, não possuindo acesso às informações que possibilitam a leitura completa de um conteúdo protegido e, portanto, mantendo a privacidade de tais conteúdos – como por exemplo os dados do usuário.

Neste mecanismo, os componentes *Clientes Privacify* e *Servidor Privacify* são confiáveis e isto significa que se algum dos dois componentes for comprometido, a privacidade dos usuários também estará comprometida. A confiabilidade do *Servidor da Rede Social* independe, pois, com a utilização do *Privacify*, as mensagens originais são substituídas por mensagens protegidas antes de serem enviadas e armazenadas pelo *Servidor da Rede Social*.

Para o mecanismo proposto neste trabalho, um conteúdo protegido é qualquer conteúdo que foi criptografado pelo *Privacify*, por exemplo, dados de localização do usuário, conversas do usuário com seus amigos da rede social, entre outros.

A proteção do conteúdo oferecida pelo *Privacify* é obtida por meio de criptografia, efetuando uma combinação entre os métodos de criptografia simétrica e assimétrica para a escrita e leitura de conteúdos privados. Além disso, oferece a possibilidade dos usuários receberem serviços personalizados. Para que este cenário seja possível, cada mensagem protegida enviada para a rede social está no formato *Privacify-Message*, ilustrada na Figura 3.2.

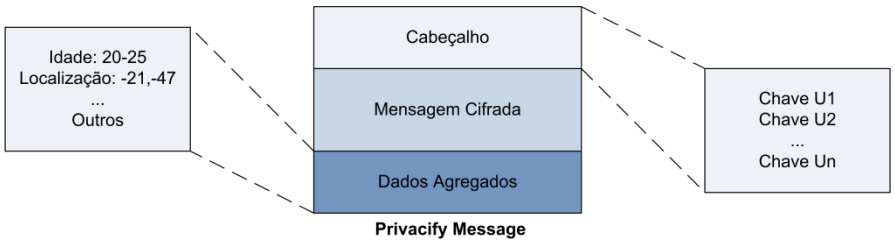


Figura 3.2 – Formato da *Privacify-Message*.

A *Privacify-Message* possui as seções *Cabeçalho*, *Mensagem Cifrada* e *Dados Agregados*. O campo *Cabeçalho* possui a listagem de destinatários e chaves

associadas para que a leitura da mensagem original possa ser feita adequadamente. A maneira como este campo é preenchido é discutida posteriormente nesta seção. O campo *Mensagem Cifrada* contém a carga útil da mensagem, em outras palavras este campo possui a mensagem original criptografada. Por fim, o campo *Dados Agregados* fornece a possibilidade das redes sociais e terceiros continuarem oferecendo serviço personalizado para os usuários. Em alguns casos não há necessidade de dados agregados e, portanto este campo é opcional. Com este campo é possível agregar algumas informações sensitivas para que os valores exatos não sejam revelados para todos. Por exemplo, o invés de fornecer a idade de um usuário precisamente, digamos 25 anos, pode-se adicionar neste campo um intervalo de idade "25-30". O formato deste campo também é discutido posteriormente nesta seção.

Cada *Cliente Privacify* possui os módulos listados na Figura 3.3. Como dito anteriormente, na prática, caso a rede social em questão seja uma aplicação *Web*, então o *Cliente Privacify* pode ser um complemento de um navegador *Web*. Para este cenário, o navegador *Web* estaria representado pelo módulo de *Acesso* e o complemento que implementa o *Privacify* estaria representado pelos módulos *Gerenciamento*, *Escrita/Leitura* e *Configurações*. Em outros casos, o módulo de *Acesso* deve ser implementado juntamente com os outros módulos do *Privacify*.

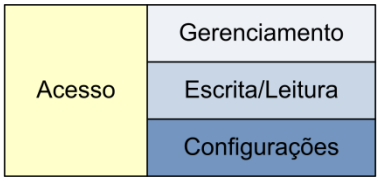


Figura 3.3 – Componentes do *Cliente Privacify*.

O módulo de *Acesso* na Figura 3.3 representa a parte do sistema que é utilizada para acessar a rede social. O módulo *Gerenciamento* é responsável por identificar o usuário ativo, carregar as configurações do usuário e carregar dados para que as mensagens protegidas possam ser lidas e escritas de maneira adequada. O módulo de *Escrita/Leitura* é responsável respectivamente por substituir as mensagens originais por mensagens protegidas antes de serem enviadas para o servidor e por ler mensagens protegidas, tornando-as mensagens que os usuários possam consumir. O módulo de *Configurações* deve permitir que o usuário ajuste

quais tipos de dados serão protegidos e o nível de proteção do *Privacify*, “baixo, médio, alto ou personalizado”. A proteção que cada nível oferece é discutida posteriormente nesta seção.

Para que a funcionalidade dos *Clientes Privacify* possa ser colocada em prática, estes devem se comunicar com o *Servidor Privacify* para a recuperação de configurações dos usuários e de outros dados. A Figura 3.4 ilustra a organização do *Servidor Privacify*.

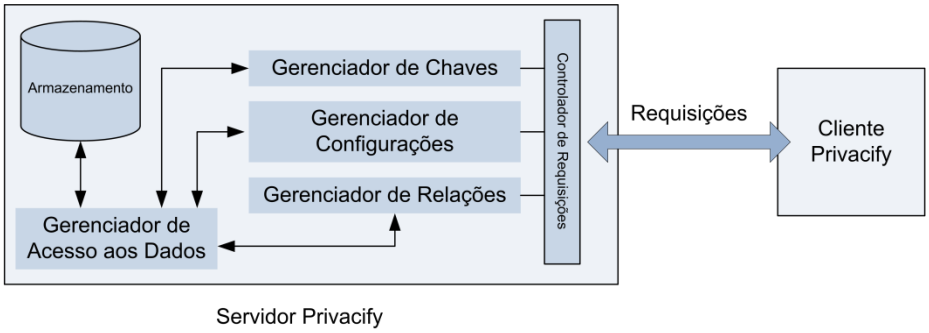


Figura 3.4 – Componentes do *Servidor Privacify*.

No lado direito da Figura 3.4, está a representação do *Cliente Privacify* que comunica-se, via um canal de comunicação seguro – por exemplo, o *HyperText Transfer Protocol Secure* (HTTPS), com o *Servidor Privacify* para obter dados do usuário ativo. Quando uma requisição chega ao servidor, é entregue ao *Controlador de Requisições* que identifica qual módulo do servidor será designado para tratar a requisição. Entre estes módulos estão o *Gerenciador de Chaves*, o *Gerenciador de Configurações* e o *Gerenciador de Relações*. Cada um destes módulos pode gravar e ler conteúdo por meio do *Gerenciador de Acesso aos Dados*.

O módulo *Gerenciador de Chaves* é responsável por registrar e recuperar a chave pública e privada do usuário ativo. O registro é feito baseando-se no identificador do usuário e nas chaves. Para efeito de segurança, a chave privada é criptografada pelo *Cliente Privacify* antes de ser enviada ao servidor. A recuperação das chaves é feita baseada no identificador do usuário ativo. O módulo *Gerenciador de Chaves* poderia ser substituído por uma autoridade certificadora externa ao mecanismo, porém optou-se por incluir este elemento ao mecanismo para torná-lo mais completo e independente.

As configurações do usuário ativo podem ser armazenadas e recuperadas pelo módulo *Gerenciador de Configurações*. Em um primeiro momento, somente informações das seções que devem ser protegidas de acordo com o nível de privacidade são tratadas por este módulo, porém em possíveis extensões do mecanismo *Privacify*, outras configurações podem ser tratadas.

O *Gerenciador de Relações* mantém informações de relações de amizade entre usuários da rede social. Para que seja possível a interação via mensagens protegidas entre os usuários na rede social, é necessário que este tipo de informação esteja em posse do *Privacify*. A relação de amizade permite que o *Servidor Privacify* recupere adequadamente as chaves públicas dos amigos de um usuário, possibilitando a troca de mensagens protegidas entre eles.

Por fim, o *Gerenciador de Acesso aos Dados* é responsável por fazer a interface entre o banco de dados e os demais módulos para que os dados sejam armazenados e recuperados de maneira correta.

Quando o usuário acessa o sistema pela primeira vez, é necessário que se faça a geração do par de chaves pública e privada, bem como a associação de seus amigos da rede social. A associação neste caso é a criação do relacionamento de amizade entre os usuários. Com este relacionamento, um usuário pode ter acesso à chave pública de seus amigos e assim pode adicioná-los como destinatário em mensagens protegidas. Este processo é ilustrado pelos passos no diagrama da Figura 3.5.

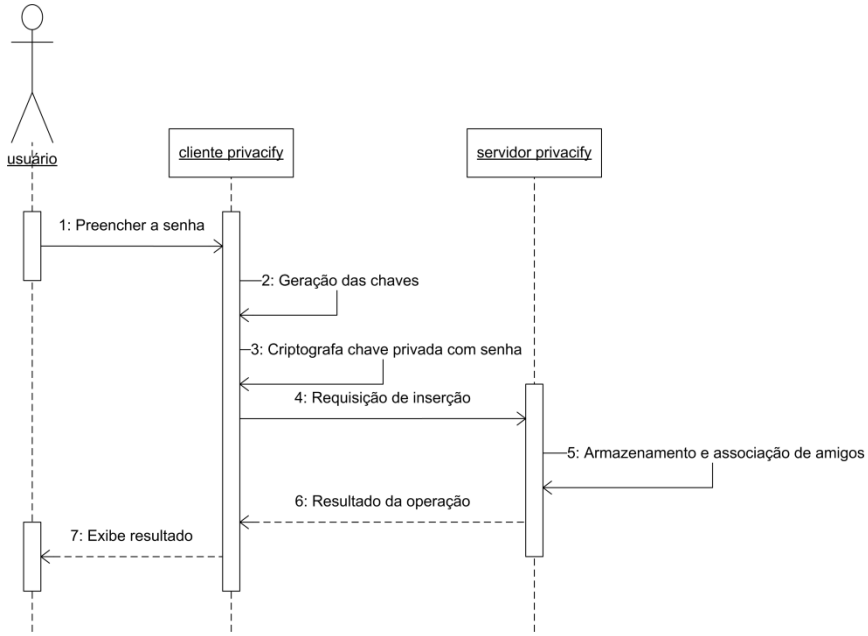


Figura 3.5 – Diagrama com os Passos para o Primeiro Acesso do Usuário.

Ao enviar uma mensagem para rede social, é solicitado ao usuário que está utilizando o *Cliente Privacify* pela primeira vez o registro de uma senha, etapa 1 apresentada na Figura 3.5. Na etapa 2, o *Cliente Privacify* gera o par de chaves pública e privada para o usuário e criptografa a chave privada com a senha fornecida pelo usuário, etapa 3. Isto previne que a chave privada seja roubada e utilizada por terceiros durante o cadastro das chaves. Em seguida, na etapa 4, é efetuada uma requisição de cadastro para o *Servidor Privacify*, enviando-se informações do usuário, dos destinatários da mensagem e do par de chaves. As informações enviadas devem ser basicamente identificadores dos usuários na rede social e parâmetros de configuração *Privacify* do usuário ativo. Na etapa 5 o *Servidor Privacify* efetua o cadastro do usuário e a associação de amigos, retornando para o *Cliente Privacify* uma mensagem de sucesso caso o cadastro for realizado sem problemas ou falha caso contrário.

A senha citada anteriormente é utilizada em dois momentos pelo sistema. Primeiramente é utilizada para criptografar a chave privada, como citado na etapa 3 da Figura 3.5. Posteriormente, a senha é utilizada para obter a chave privada, que permite a leitura de mensagens destinadas ao usuário ativo. Esta senha é criada pelo próprio usuário, não fica armazenada no *Servidor Privacify* e deve ser lembrada a cada acesso. Uma forma de esta senha ser utilizada tanto para criptografar a

chave privada quanto para obtê-la posteriormente é a utilização do método de criptografia simétrico, que é o método proposto para a utilização do *Privacify*.

Os acessos sucessivos à rede social com o *Cliente Privacify* são ilustrados na Figura 3.6. No momento em que o usuário inicia a interação com a rede social, é solicitada a senha que foi utilizada para o primeiro acesso, etapa 1 da Figura 3.6. Em seguida, na etapa 2, o *Cliente Privacify* faz a requisição das chaves do usuário para o *Servidor Privacify*. Na etapa 3, para aumentar a segurança, ao invés de enviar as chaves no primeiro momento, o servidor gera um desafio baseando-se na chave pública do usuário em questão e envia este desafio juntamente com a chave privada criptografada. Então, o *Cliente Privacify* utiliza a senha e um método de criptografia simétrica para obter a chave privada, que é utilizada para resolver o desafio. Neste momento, etapa 4, o desafio resolvido é enviado para o servidor, que envia os dados do usuário caso o desafio resolvido esteja correto ou então envia uma mensagem de erro, etapa 5. Caso o desafio tenha sido resolvido com sucesso, as chaves são armazenadas localmente até o final da sessão e uma mensagem de sucesso é exibida para o usuário, etapa 6. Além das chaves do usuário, informações adicionais também são enviadas pelo servidor e armazenadas localmente caso o desafio seja resolvido com sucesso, como é o caso da relação de amigos e configurações de privacidade. Caso o desafio não seja resolvido corretamente, uma mensagem de erro é exibida e a senha é solicitada novamente.

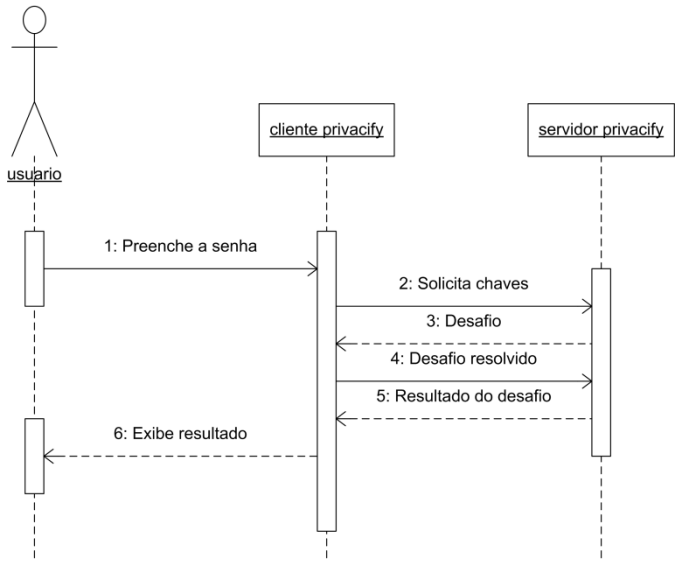


Figura 3.6 – Diagrama com os Passos para Acessos Sucessivos.

Após as chaves terem sido recuperadas, o usuário está pronto para ler e enviar mensagens protegidas com a utilização do *Privacify*. Este tipo de interação, envio de mensagens, é ilustrado pelo diagrama da Figura 3.7. Primeiramente, o usuário escolhe um ou mais destinatários da mensagem e em seguida preenche a mensagem que será enviada, etapas 1 e 2 da Figura 3.7. Em seguida, o *Cliente Privacify* gera um segredo aleatório, etapa 3, que será utilizado para criptografar a mensagem original via criptografia de chave simétrica. Na etapa 4, a *Privacify-Message* é montada. Primeiramente, a mensagem original é criptografada com base no segredo gerado aleatoriamente e o resultado desta operação é adicionado no campo *Mensagem Cifrada* da *Privacify-Message*. Após isto, o segredo aleatório é criptografado, via criptografia de chave assimétrica, com a chave pública de cada destinatário e o resultado de cada operação juntamente com o identificador do destinatário é adicionado no campo *Cabeçalho* da *Privacify-Message*. Além das chaves públicas dos destinatários, a chave pública do usuário ativo também passa pelo mesmo processo, para que seja possível ao usuário ativo ler a mensagem no futuro. Caso haja dados agregados, estes são adicionados à mensagem, etapa 5.

Após estas etapas, a *Privacify-Message* está composta e pode ser enviada para armazenamento, substituindo a mensagem original. Então, o *Cliente Privacify* envia a *Privacify-Message* para que o armazenamento seja feito pela rede social, etapa 6 da Figura 3.7. Para o usuário, estes passos devem ser transparentes de forma que ele utilize a rede social da mesma maneira que utilizava sem o *Privacify*. Certamente, algumas situações irão requerer intervenção do usuário, por exemplo, a configuração do nível de privacidade que o *Privacify* deverá desempenhar, porém o mecanismo tenta minimizar o número de intervenções na experiência do usuário. As etapas 7 e 8 exibem a resposta do servidor da rede social para o usuário.

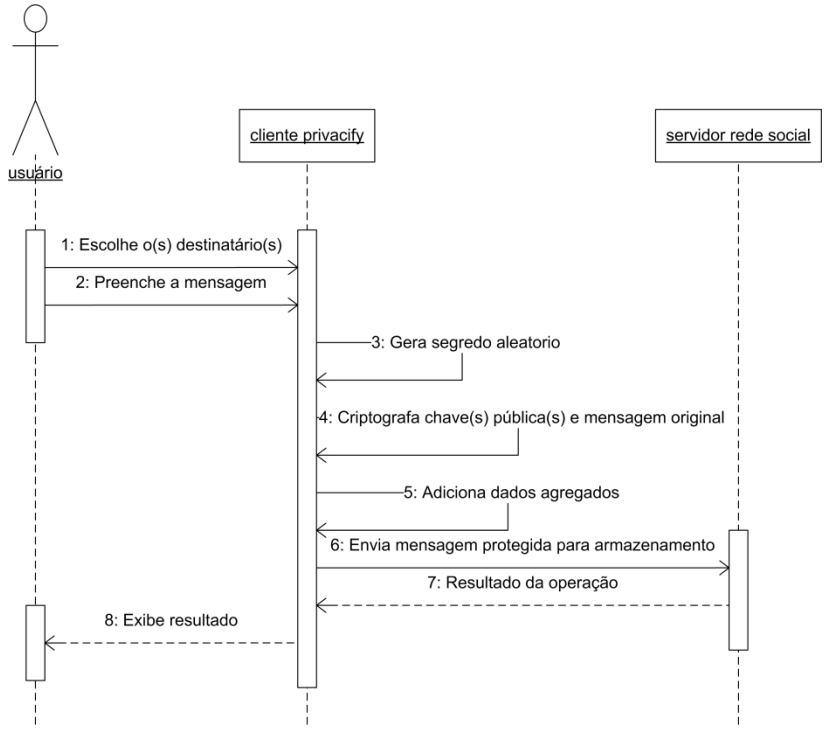


Figura 3.7 – Diagrama com os Passos para o Envio de Mensagem.

Assim como a escrita da mensagem, o processo de leitura também é feito de forma transparente para o usuário, sendo que o *Cliente Privacify* encarrega-se de identificar o conteúdo protegido e transformá-lo em um formato que possa ser consumido pelo usuário, a Figura 3.8 ilustra as etapas envolvidas no processo de leitura da mensagem.

Nas etapas 1 e 2 da Figura 3.8, o usuário solicita algum conteúdo por meio do *Cliente Privacify*. Ao obter o conteúdo solicitado ao *Servidor da Rede Social*, etapa 3, o *Cliente Privacify* verifica a existência de seções protegidas no conteúdo, etapa 4. Caso encontre alguma seção protegida, então é verificado se o identificador do usuário ativo encontra-se no *Cabeçalho* da mensagem, para que a *Mensagem Cifrada* possa ser lida. Se o identificador for encontrado, significa que o usuário ativo é um dos destinatários da mensagem e então o *Cliente Privacify* pode utilizar a chave privada do usuário ativo para obter o segredo que permite a leitura da *Mensagem Cifrada*. De posse do segredo, a mensagem original é lida a partir da *Mensagem Cifrada*, via criptografia de chave simétrica, e exibida para o usuário. Caso o *Cliente Privacify* não encontre o identificador do usuário no *Cabeçalho*, uma mensagem é exibida ao usuário para informar que certos conteúdos são protegidos e que ele não está relacionado na lista de usuários que podem ler tal conteúdo.

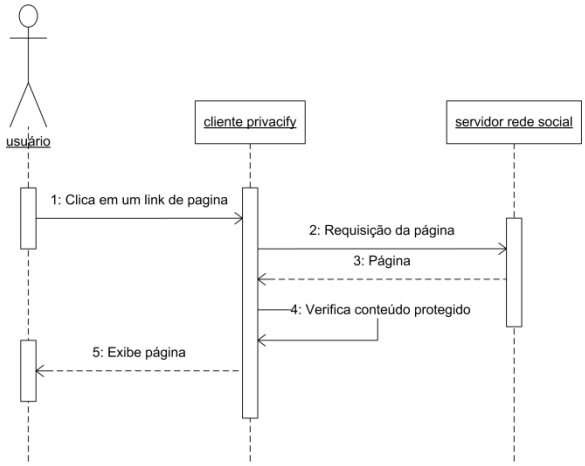


Figura 3.8 – Diagrama com os Passos para a Leitura de Mensagem.

O mecanismo proposto combina a criptografia com uma estrutura de mensagem e permite que a privacidade do usuário seja mantida em alguns aspectos, possibilitando mesmo assim a obtenção de serviços personalizados. Porém, não garante a privacidade total do usuário e pode-se dizer que o mecanismo permite estender a privacidade fornecida pela rede social. Os níveis de privacidade oferecidos pelo mecanismo estão ilustrados na Figura 3.9.

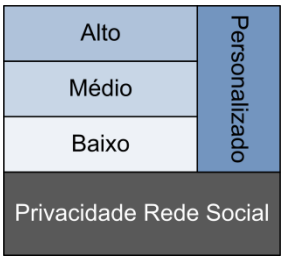


Figura 3.9 – Níveis de Privacidade.

Na Figura 3.9, o nível mais baixo é a privacidade oferecida pela própria *Rede Social*. Este nível de privacidade permite que o usuário ativo controle quais usuários ou grupo de usuários da rede social podem acessar e utilizar certos tipos de informações. No entanto, geralmente não permite que o mesmo tipo de configuração seja feito em relação à própria rede social e a terceiros. Os níveis de privacidade *Baixo*, *Médio*, *Alto* e *Personalizado* são oferecidos pelo *Privacify* e estão diretamente relacionados com a seção *Dados Agregados* da *Privacify-Message*.

De maneira geral, os níveis de privacidade regulam a precisão dos dados agregados. Isto significa que para o nível de privacidade “Baixo”, a precisão é maior

que para o nível de privacidade “Alto”. Porém, os dados agregados sempre possuem dados menos precisos em relação aos dados originais. Ao definir o nível de privacidade como “Baixo”, “Médio” ou “Alto”, todas as informações que possuem *Dados Agregados* serão configuradas para o mesmo nível. Para associar níveis de privacidade distintos a diferentes tipos de *Dados Agregados*, pode-se utilizar o nível de privacidade “Personalizado”, que permite a escolha do nível de privacidade individual de cada item a ser protegido.

Desta forma, o usuário que utiliza serviços personalizados na rede social deve notar que quanto maior o nível de privacidade, menor a precisão devido a confusão dos dados, o que pode implicar na obtenção de serviços personalizados com qualidade inferior ao desejado. Isto significa que o nível de privacidade “Baixo” aplica um baixo ajuste de precisão ao valor agregado, tornado os dados agregados próximo aos dados originais, enquanto o nível de privacidade “Alta” aplica um ajuste de precisão maior.

O formato do campo *Dados Agregados* é definido como sendo um par tipo-valor, que é bem conhecido para que aplicações de terceiros possam utilizá-lo. O primeiro item do par é o **`dataType`**, que identifica o tipo do dado, por exemplo, um valor numérico, um intervalo de valores, um texto, entre outros. O segundo item é o **`dataValue`**, que é o conteúdo disposto conforme o **`dataType`**, na forma de dados agregados.

Embora as informações dos dados agregados possam ser lidas por todos, a semântica da *Privacify-Message* é definida pela rede social. Por exemplo, suponha que uma mensagem protegida que descreve o local em que um usuário está morando atualmente por meio de sua latitude e longitude. O campo *Mensagem Cifrada* contém a latitude e longitude originais, porém, de maneira criptografada e o campo *Dados Agregados* contém **`dataType`** com o valor *localização* e **`dataValue`** com o valor de latitude e longitude desviado. Este valor desviado é baseado na latitude e longitude originais com um ajuste de precisão de acordo com o nível de privacidade. Uma aplicação de terceiro que ler o conteúdo fornecido pela rede social, a *Privacify-Message*, será capaz de identificar que o valor latitude e longitude no campo **`dataValue`** se refere ao local onde o usuário mora e assim fornecer o serviço personalizado ao usuário de acordo com a precisão ajustada no nível de privacidade. Por outro lado, aplicações que acessarem a *Privacify-Message* fora do contexto da rede social, terão que descobrir a semântica da mensagem para então

poder utilizar o valor contido nos *Dados Agregados*. Um estudo de caso com a aplicação dos *Dados Agregados* é apresentado no Capítulo 4.

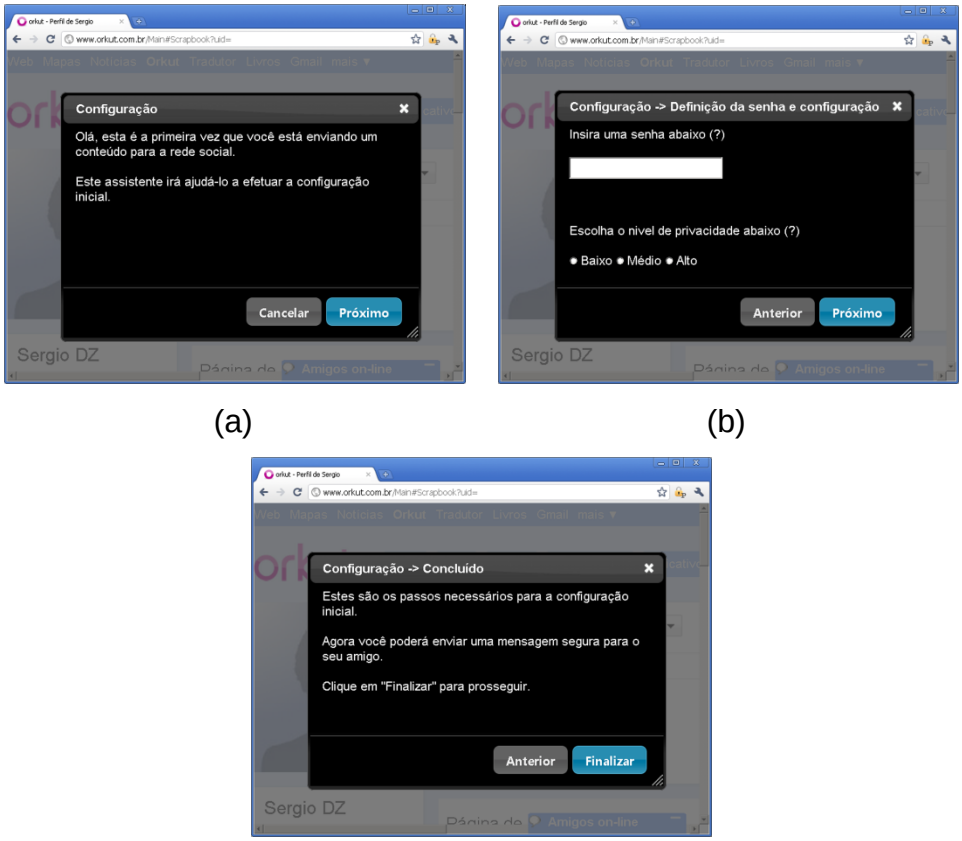
3.4 Implementação

O mecanismo proposto na seção 3.3 possui três elementos principais quando o foco é a implementação. Estes três elementos são a *Rede Social*, o *Cliente Privacify* e o *Servidor Privacify*. Para a implementação prova de conceito, Orkut foi escolhido como a *Rede Social*, pela sua popularidade no Brasil (COMSCORE, 2010), o navegador da *Web Google Chrome*[16], juntamente com uma extensão que implementa o *Privacify* formam o *Cliente Privacify* e um serviço RESTful (FIELDING, 2000; RICHARDSON; RUBY, 2007) foi implementado para responder as requisições de chave e configurações de privacidade do usuário, *Servidor Privacify*.

Esta seção, primeiramente, mostra algumas telas do *Cliente Privacify* e o fluxo em que algumas etapas ocorrem e, posteriormente, apresenta alguns detalhes de implementação tanto do *Cliente Privacify* quanto do *Servidor Privacify*. A partir deste ponto, iremos usar o termo *Privacify-C* para referenciar o *Cliente Privacify* e *Privacify-S* para referenciar o *Servidor Privacify*.

Primeiramente, para iniciar a utilização do *Privacify* o usuário deve instalar o navegador *Google Chrome* e em seguida instalar a extensão *Privacify-C* no navegador. Em seguida, o usuário acessa a rede social normalmente e na primeira vez que for enviar um conteúdo para a rede social o *Privacify-C* faz uma intervenção, para que o usuário possa inserir uma senha e opcionalmente configurar o nível de privacidade. A Figura 3.10 exibe os passos que guiam o usuário até a conclusão desta etapa.

[16] Disponível em: <http://www.google.com/chrome/?hl=pt-BR>. Acesso em: 10/04/2011.



(a) (b)

(c)

Figura 3.10 – Configuração Inicial do *Privacify-C*.

Após concluir o passo a passo exibido na Figura 3.10, o usuário ativo pode enviar mensagens protegidas caso o usuário alvo já esteja utilizando o *Privacify*. Caso contrário, o envio da mensagem protegida é suspenso e o usuário é avisado sobre este requisito. O par de chaves pública e privada do usuário ativo é gerado no momento em que o passo a passo é avançado da etapa 2 para a etapa 3, respectivamente Figura 3.10 (a) e Figura 3.10 (b), e após esta etapa, o par de chaves é enviado para o servidor juntamente com a configuração do nível de privacidade. Embora o nível de privacidade “Personalizado” não apareça na Figura 3.10 (b), é possível configurá-lo em uma etapa posterior, conforme mostrado na Figura 3.12. Também nesta etapa, a relação de amizade é importada para o *Privacify-S*, permitindo que o usuário comunique-se, via mensagens protegidas, com os amigos da rede social que já estão utilizando o *Privacify*.

A comunicação com mensagens protegidas é ilustrada na Figura 3.11. A Figura em questão ilustra o envio de mensagem protegida por um usuário que não configurou o nível de privacidade em que o *Privacify* deve atuar e neste caso, ao

enviar qualquer conteúdo o usuário é perguntado se o conteúdo deve ser enviado em sua forma original ou na forma protegida.

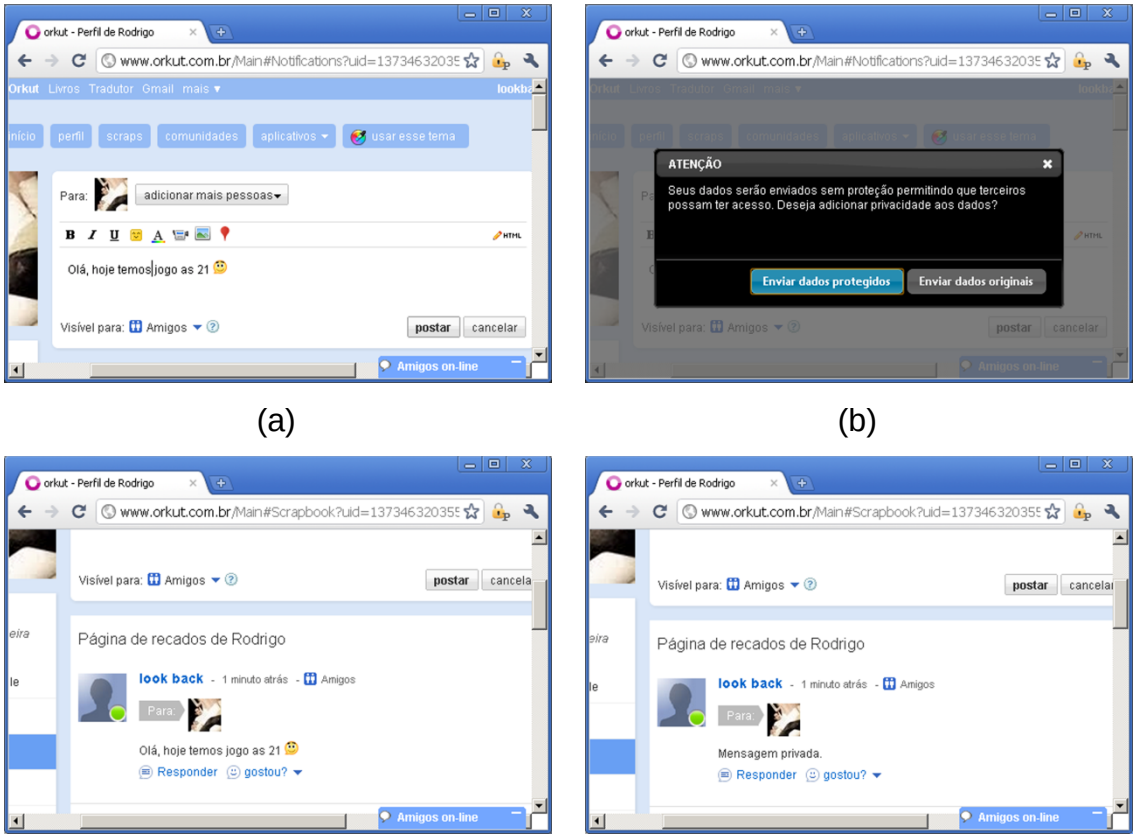
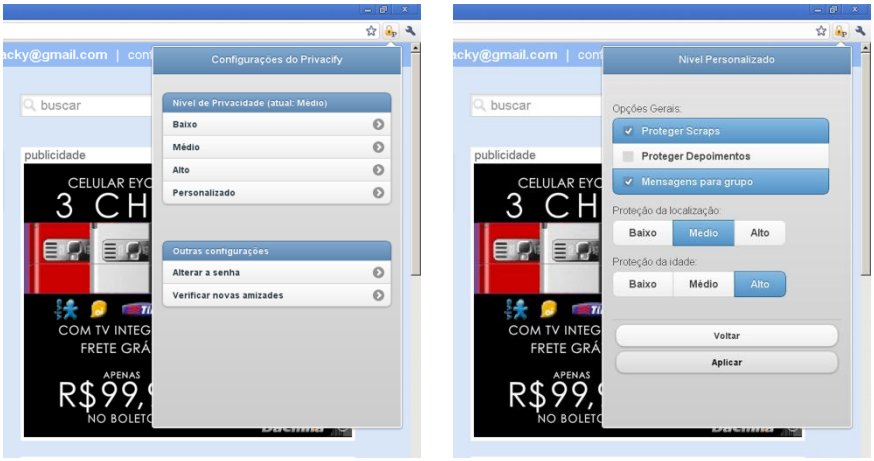


Figura 3.11 – Enviando e Visualizando uma Mensagem pelo *Privacify-C*.

Na Figura 3.11 (a) o usuário ativo compõe uma mensagem e em seguida clica no botão “Postar”. O *Privacify-C* verifica o nível de privacidade preferido pelo usuário ativo e exibe a janela de confirmação, Figura 3.11 (b). Caso o usuário ativo opte por enviar a mensagem de maneira protegida, o *Privacify-C* gera a *Privacify-Message* e efetua o envio para a *Rede Social*. Após esta etapa, o usuário destino pode visualizar a mensagem original, Figura 3.11 (c), e um usuário que não está relacionado no *Cabeçalho* da *Privacify-Message*, visualiza uma mensagem padrão – “Mensagem privada”, Figura 3.11 (d).

O usuário ainda pode alterar as configurações mesmo após ter preenchido a senha e o nível de privacidade no passo a passo exibido na Figura 3.10. Para tal tarefa, basta acionar o botão do *Privacify-C* na barra do navegador para que seja aberta uma janela com as opções de configuração. A Figura 3.12 (a) exibe a tela principal de configurações do *Privacify-C*. Os itens “nível de privacidade” e “senha”

permitem alterar o nível de privacidade e a senha, respectivamente. O item “Verificar novas amizades” permite que a relação de amizade seja atualizada no *Privacify-S*, tornando possível que o usuário ativo envie conteúdo protegido para todos amigos que também utilizam o *Privacify*. A Figura 3.12 (b) ilustra a tela seguinte que é exibida quando o usuário ativo seleciona o nível de privacidade personalizado. Nesta tela o usuário pode configurar individualmente os níveis dos itens que deseja proteger.



(a) (b)

Figura 3.12 – Configurações do *Privacify-C*.

Após ter apresentado algumas das possibilidades de utilização do *Privacify-C*, pode-se entender alguns detalhes de implementação que são apresentados a seguir. Por ser uma extensão do *Google Chrome*, o desenvolvimento do *Privacify-C* é baseado na linguagem *Javascript*, que é interpretada pelo próprio navegador. Além disto, algumas bibliotecas *Javascript* de terceiros foram utilizadas para o desenvolvimento das interfaces gráficas do usuário e para os procedimentos de criptografia. Do lado do servidor, o *Privacify-S* é implementado na linguagem PHP com a utilização do banco de dados MySQL.

O funcionamento básico do *Privacify-C* é obtido por meio da manipulação dos objetos *HyperText Markup Language* (HTML) presentes na página via *Document Object Model* (DOM)[17]. Por exemplo, ao acessar a rede social o usuário corrente

[17] Disponível em: <http://www.w3.org/DOM/>. Acesso em: 10/04/2011.

necessita dos dados de chaves e configurações pessoais do *Privacify*. Para isto, o *Privacify-C* identifica o usuário ativo conforme o código da Figura 3.13.

```
1  function getUserKey() {
2      try {
3          if( currentUserUID ) return;
4
5          var result = document.evaluate( "//a", document, null, 0, null );
6          var marker = "Main#Profile?uid=";
7          var item;
8          var href;
9          var index;
10         while( item = result.iterateNext() ) {
11             href = item.href;
12             index = href.indexOf( marker );
13             if( index >= 0 ) {
14                 index += marker.length;
15                 currentUserUID = href.substring( index, href.length );
16                 retrieveUserData( currentUserUID, decryptChallenge );
17                 break;
18             }
19         }
20     } catch( error ) {
21         if( debugErrorsOn ) {
22             alert( error );
23         }
24     }
25 }
```

Figura 3.13 – Trecho de Código para Identificação do Usuário Ativo.

A linha 1 da Figura 3.13 inicia a declaração da função **`getUserKey`** que identifica o usuário ativo. A variável **`currentUserUID`**, linha 3, é de contexto global e armazena o identificador único do usuário ativo que está utilizando o *Privacify-C*. Este identificador é o mesmo gerado pelo Orkut. Ainda na linha 3, é verificado se o usuário corrente já foi identificado e em caso positivo o restante da função é simplesmente ignorado. Caso contrário, percorre-se a página HTML em busca do identificador do usuário. A linha 5 contém uma instrução que avalia expressões *XML Path* (XPath)[18] em um documento *Extensible Markup Language* (XML) e o resultado, armazenado na variável `result`, é um nó ou um conjunto de nós que correspondem à expressão XPath. Neste contexto, estamos procurando por todas as tags de link HTML na página corrente. As linhas 6, 7, 8 e 9 declaram variáveis auxiliares utilizadas para percorrer o resultado da variável **`result`**. O intervalo de linhas de 10 à 19 implementa a iteração sobre o resultado de tags de link na página corrente. A variável **`href`** na linha 11 armazena a *Uniform Resource Locator* (URL) do link,

[18] Disponível em: <http://www.w3.org/TR/xpath/>. Acesso em: 10/04/2011.

enquanto a variável **`index`** na linha 12 é utilizada para verificar se trata-se do link que direciona para a página principal do usuário, linha 13. Caso seja o link para a página inicial do usuário, então o identificador do usuário é armazenado na variável **`currentUserUID`** na linha 15 e a solicitação de dados do usuário é requisitada para o *Privacify-S*, linha 16. Como esta é uma requisição assíncrona, uma função de retorno é passada como parâmetro para que o *Privacify-C* possa manipular a resposta do *Privacify-S*.

Antes que os dados do usuário sejam retornados para o *Privacify-C*, o *Privacify-S* emite um desafio para verificar a procedência de requisição. Desta forma, a requisição de dados do usuário é tratada inicialmente pelo trecho de código escrito em PHP exibido na Figura 3.14. Antes que o conteúdo da Figura 3.14 seja explicado, é importante notar que para efeito de implementação prova de conceito, neste trabalho não são armazenadas as chaves pública e privada propriamente ditas. Sabe-se que a chave privada é formada pelo módulo *Rivest, Shamir e Adleman* (RSA) e expoente de chave secreta, os números n e d respectivamente, e que a chave pública é formada pelo módulo RSA e o expoente de chave pública, n e e respectivamente (RIVEST *et al.*, 1978). Então, ao invés de armazenar a chave pública e privada, apenas os números n e d são armazenados separadamente pelo *Privacify-S*, porém número d é armazenado de maneira criptografada. O número e não é armazenado em banco de dados e trata-se de um valor que fica embutido junto com a configuração da aplicação, sendo que este valor é comum tanto no *Privacify-C* quanto no *Privacify-S*. Desta forma, ao dizer que a chave privada é criptografada, estamos na verdade dizendo que o expoente de chave secreta é criptografado e isto facilita a compreensão do texto.

Na Figura 3.14, a linha 1 inicia o script PHP e a linha 2 inclui o arquivo de cabeçalho. Este arquivo contém definições de funções, informações de banco de dados e gerenciamento de sessão. Nas linhas 3 e 4 o identificador do usuário que está fazendo a requisição é identificado e verificado, respectivamente. Caso o identificador não esteja vazio, então é gerado um desafio aleatório, linha 5, que dependerá basicamente da senha do usuário para ser resolvido. O retorno da função **`generateRandomChallenge`** pode ser nulo em caso de falha na geração do desafio ou então um vetor associativo que contém o desafio aleatório criptografado com a chave pública do usuário, a chave privada que estava armazenada no banco de

dados de maneira criptografada e a chave pública do usuário que está efetuando a requisição. Além disto, a função **`generateRandomChallenge`** ainda registra o desafio aleatório original em sessão para que este possa ser comparado com o desafio resolvido pelo *Privacify-C*. A linha 6 verifica se o desafio foi gerado com sucesso e em caso afirmativo uma mensagem no formato *JavaScript Object Notation* (JSON)[19] é enviada com o desafio e as chaves, linhas 7 a 16. Os campos **`status`** e **`service`**, linhas 9 e 10 respectivamente, são metadados da mensagem e permitem que o *Privacify-C* trate as respostas de maneira adequada. Caso a geração do desafio falhe ou então a identificação do usuário não possa ser feita, linhas 18 a 22 e 25 a 29 respectivamente, uma mensagem de erro é retornada.

```
1  <?php
2  include_once "includes/header.php";
3  $uid = getCurrentUID();
4  if( !empty( $uid ) ) {
5      $challenge = generateRandomChallenge( $uid );
6      if( !empty( $challenge ) ) {
7          echo json_encode(
8              array(
9                  'status' => STATUS_SUCCESS,
10                 'service' => 'retrieve-challenge',
11                 'body' => array(
12                     'challenge' => $challenge[ 'cipher_challenge' ],
13                     'rsa_d' => $challenge[ 'rsa_d' ],
14                     'rsa_n' => $challenge[ 'rsa_n' ],
15                 )
16             ) );
17     } else {
18         echo json_encode( array(
19                                 'status' => STATUS_ERROR,
20                                 'service' => 'retrieve-challenge',
21                                 'body' => 'Parâmetros inválidos.'
22                             ) );
23     }
24 } else {
25     echo json_encode( array(
26                             'status' => STATUS_ERROR,
27                             'service' => 'retrieve-challenge',
28                             'body' => 'Parâmetros inválidos.'
29                         ) );
30 }
31 include_once "includes/footer.php";
32 ?>
```

Figura 3.14 – Trecho de Código para Geração do Desafio Aleatório.

A resolução do desafio emitido pelo *Privacify-S* é efetuada sem a intervenção do usuário. A Figura 3.15 exibe o trecho de código do *Privacify-C* que é responsável

[19] Disponível em: <http://www.json.org>. Acesso em: 12/04/2011.

por resolver o desafio. A linha 1 contém a declaração da função. Na linha 3, a resposta do *Privacify-S* é convertida para um objeto *Javascript* por meio de métodos da biblioteca JSON. Na linha 4, é obtida uma referência para a chave privada criptografada do usuário. A chave privada é obtida na linha 5 por meio do algoritmo de criptografia de chave simétrica *Advanced Encryption Standard* (AES), cuja implementação utilizada neste trabalho é a *Stanford Javascript Crypto Library* (SJCL)[20]. Assim, os parâmetros passados para obter a chave privada na linha 5, são a senha do usuário ativo e a chave privada criptografada. Para a resolução do desafio é utilizado o algoritmo de chave assimétrica RSA e a implementação utilizada neste trabalho é a biblioteca Javascript JSBN[21]. Na linha 7, um objeto de chave RSA é construído e na linha 8 a chave privada é inicializada. A linha 10 declara a variável **`resolved`** que contém o desafio resolvido. Por fim, as linhas 11 a 15 enviam uma requisição para o Privacify-S com o identificador do usuário e o desafio resolvido. Como esta também é uma requisição assíncrona, é registrado um método de retorno para tratar a resposta do servidor.

```
1  function decryptChallenge( param ) {
2      try {
3          var challenge   = JSON.parse( param );
4          var cipherRSA_D = challenge.body.rsa_d;
5          var rsa_d       = sjcl.decrypt( currentPWD, cipherRSA_D );
6
7          var key = new RSAKey();
8          key.setPrivate( challenge.body.rsa_n, defaultE, rsa_d );
9
10         var resolved = key.decrypt( challenge.body.cipher_challenge );
11         retrieveUserDataChallengeResolved(
12             currentUserUID,
13             resolved,
14             onUserDataReady
15         );
16     } catch( error ) {
17         if( debugErrorsOn ) {
18             alert( error );
19         }
20     }
21 }
```

Figura 3.15 – Trecho de Código para Resolver o Desafio Enviado pelo *Privacify-S*.

Para verificar se o desafio foi resolvido corretamente, o *Privacify-S* implementa o trecho de código exibido na Figura 3.16. Assim como exibido na Figura

[20] http://crypto.stanford.edu/sjcl/

[21] http://www-cs-students.stanford.edu/~tjw/jsbn/

3.14, as linhas 1, 2 e 3 contêm a inicialização do script PHP, a inclusão do arquivo de cabeçalho e o armazenamento do identificador único do usuário que está efetuando a requisição na variável **`$uid`**. Na linha 4, o identificador do usuário é passado como parâmetro para a função **`getChallengeResponse`** que, basicamente, obtém o desafio resolvido do usuário e o compara com o desafio emitido anteriormente, retornando verdadeiro, caso sejam iguais, ou falso, caso contrário. Em seguida, na linha 5, é verificado se o identificador do usuário é valido e se a resposta está correta. Caso ambos estejam corretos, os dados do usuário são obtidos na linha 6 e armazenados na variável **`$user`**. Os dados do usuário são os números que compõem o par de chaves pública e privada, dados de preferência do usuário e uma lista de amigos que contêm o par identificador do amigo e chave pública. Caso os dados do usuário forem recuperados corretamente, linha 7, então uma mensagem no formato JSON é retornada para o *Privacify-C*, linhas 8 a 16. As linhas 18 a 21 e 24 a 26 são, respectivamente, mensagens de erro enviadas caso os dados do usuário não sejam encontrados, ou caso a identificação do usuário não possa ser feita.

```
1  <?php
2  include_once "includes/header.php";
3  $uid      = getCurrentUID();
4  $response = getChallengeResponse( $uid );
5  if( !empty( $uid ) && $response ) {
6      $user = getUserData( $uid );
7      if( !empty( $user ) ) {
8          echo json_encode(
9                  array(
10                     'status' => STATUS_SUCCESS,
11                     'service' => 'retrieve',
12                     'body' => array(
13                         'userInfo' => $user[ 'user_info' ],
14                         'friendsInfo' => $user[ 'friends_info' ]
15                     )
16                 ) );
17     } else {
18         echo json_encode( array( 'status' => STATUS_ERROR,
19                                  'service' => 'retrieve',
20                                  'body' => 'Parâmetros inválidos.'
21                                ) );
22     }
23 } else {
24     echo json_encode( array( 'status' => STATUS_ERROR,
25                              'service' => 'retrieve',
26                              'body' => 'Parâmetros inválidos.' ) );
27 }
28 include_once "includes/footer.php";
29 ?>
```

Figura 3.16 – Trecho de Código para Enviar Dados do Usuário.

A última etapa para obter os dados do usuário é exibida na Figura 3.17.

```
1  function onUserDataReady( param ) {
2      try {
3          if( param ) {
4              var obj = JSON.parse( param );
5              if( obj.status == STATUS_SUCCESS ) {
6                  var cipherRSA_D = obj.body.userInfo.rsa_d;
7                  var rsa_d       = sjcl.decrypt( currentPWD, cipherRSA_D );
8                  var rsa_n       = obj.body.userInfo.rsa_n;
9
10                 key = new RSAKey();
11                 key.setPublic( rsa_n, defaultE );
12                 key.setPrivate( rsa_n, defaultE, rsa_d );
13
14                 friends = {};
15                 var returnedFriends = obj.body.friendsInfo;
16                 var length = returnedFriends.length;
17                 var friend;
18                 var i;
19                 for( i = 0; i < length; i++ ) {
20                     friend = returnedFriends[ i ];
21                     friends[ friend.friend_uid ] = friend.friend_rsa_n;
22                 }
23             }
24         }
25     } catch( error ) {
26         if( debugErrorsOn ) {
27             alert( error );
28         }
29     }
30 }
```

Figura 3.17 – Trecho de Código para Armazenamento das Chaves.

Inicialmente, na linha 1, é declarada a função **`onUserDataReady`**, que manipula a resposta do *Privacify-S* para inicializar a estrutura de dados do usuário. A resposta do servidor é verificada em dois momentos. Primeiramente, é verificado se a mensagem de resposta existe, linha 3, e posteriormente é verificado se os dados do usuário foram recuperados corretamente, linha 5. Entre estas duas verificações, linha 4, um objeto Javascript que representa a resposta do servidor é criado e armazenado na variável **`obj`**. As linhas 6, 7 e 8 contêm declarações de variáveis que armazenam respectivamente o número d criptografado, o número d e o número n do usuário ativo. Na linha 10, a variável **`key`**, de contexto global, é construída e em seguida inicializada com a chave pública e privada do usuário ativo, nas linhas 11 e 12, respectivamente. A linha 14 inicializa a variável de contexto global **`friends`**, que é uma estrutura para armazenar a lista de amigos do usuário ativo, contendo o par identificador único do amigo e sua chave pública, sendo indexada pelo identificador único do amigo. A linha 15 atribui a lista de informações de amigos à variável **`returnedFriends`**, que é utilizada para popular a variável **`friends`**. Na linha 16, o

número de amigos retornados é obtido e, nas linhas 17 e 18, variáveis auxiliares para iterar sobre a lista de amigos são declaradas. Por fim, a lista de amigos é iterada adicionando-se o identificador único do usuário e a chave pública correspondente na variável **`friends`**, linhas 19 a 22.

Por fim, as operações de escrita e leitura de mensagem protegida são apresentadas nas Figuras 3.18, 3.19 e 3.20.

```
1  function encrypt( originalMessage, aggregateData, uidsTo ) {
2      try {
3          if( key ) {
4              var length = uidsTo.length;
5              var header = {};
6              var uid;
7              var password = generateRandomPassword();
8              var friendKey = new RSAKey();
9              for( var i = 0; i < length; i++ ) {
10                 uid = uidsTo[ i ];
11                 if( friends[ uid ] ) {
12                     friendKey.setPublic( friends[ uid ], "10001" );
13                     header[ uid ] = friendKey.encrypt( password );
14                 }
15             }
16             header[ currentUserUID ] = key.encrypt( password );
17
18             var cipherMessage    = sjcl.encrypt( password, originalMessage );
19             var privacifyMessage = composePrivacifyMessage(
20                 header, cipherMessage, aggregateData
21             );
22             postMessage( privacifyMessage );
23         }
24     } catch( error ) {
25         if( debugErrorsOn ) {
26             alert( error );
27         }
28     }
29 }
```

Figura 3.18 – Trecho de Código para Enviar uma Mensagem Protegida.

Na Figura 3.18 é exibido o trecho de código para enviar uma mensagem protegida. Inicialmente, na linha 1, é declarada a função **`encrypt`**, que recebe como parâmetro a mensagem original, **`originalMessage`**, os dados agregados, **`aggregateData`**, e uma lista com identificadores de usuários que serão adicionados como destinatários da mensagem, sendo que o parâmetro **`aggregateData`** é opcional. Na linha 3 é verificado se a variável global **`key`** foi inicializada e em caso afirmativo, a *Privacify-Message* começa a ser composta. Nas linhas 4, 5 e 6 são declaradas algumas variáveis auxiliares para a montagem do cabeçalho da mensagem. Na linha 7 é gerado um segredo aleatório que é utilizado para cifrar a

mensagem original. O mesmo segredo aleatório é criptografado com a chave pública de cada usuário incluindo o remetente da mensagem, código das linhas 8 a 16. A linha 8 cria um objeto que é utilizado para criptografar o segredo aleatório com a chave pública de cada destinatário. Na linha 9, o laço para iterar sobre os destinatários é iniciado. Uma referência para o identificador de cada usuário é obtida na linha 10 e, em seguida, na linha 11, é verificado se as informações do usuário em questão estão disponíveis. Caso estejam, a chave pública do usuário em questão é inicializada e uma entrada no cabeçalho é criada com o identificador do usuário e o segredo aleatório criptografado, linhas 12 e 13. A linha 16 adiciona informações do usuário ativo no cabeçalho da mensagem. Em seguida, a mensagem original é criptografada na linha 18, tomando como parâmetro o segredo aleatório gerado na linha 7. As linhas 19, 20 e 21 são responsáveis pela geração da *Privacify-Message* por meio da função **`composePrivacifyMessage`**, que recebe os parâmetros **`header`**, **`cipherMessage`** e **`aggregateData`**. Por fim, a mensagem é postada na rede social, linha 22.

As Figuras 3.19 e 3.20 mostram o trecho de código para que uma página com conteúdo protegido possa ser lida pelo usuário. O trecho de código do *Privacify-C* na Figura 3.19, busca por conteúdos protegidos em uma página da rede social, enquanto o trecho de código da Figura 3.20 converte a mensagem para um formato que o usuário possa consumir.

```
1  function doDecryptPage() {
2      try {
3          var result = document.evaluate( "//span", document, null, 0, null );
4          var item;
5          var i = 0;
6          while( item = result.iterateNext() ) {
7              if( hasPrivacifyMessage( item ) ) {
8                  decrypt( item, getPrivacifyMessage( item )  );
9              }
10         }
11     } catch( error ) {
12         if( debugErrorsOn ) {
13             alert( error );
14         }
15     }
16 }
```

Figura 3.19 – Trecho de Código para Leitura de uma Página Criptografada.

A linha 1 da Figura 3.19 faz a declaração da função **`doDecryptPage`**. Na linha 3, uma lista de elementos **`span`** é retornada. Estes são os elementos HTML

responsáveis por exibir as mensagens na página HTML da rede social. Nas linhas 4 e 5 são declaradas variáveis auxiliares para percorrer a lista de elementos **`span`**. Na linha 6 é iniciado o laço para iterar os elementos **`span`** e na linha 7 é verificado se o item contém conteúdo protegido. Caso contenha, a função **`decrypt`** é chamada, passando-se como parâmetro o elemento **`span`** e a *Privacify-Message*, linha 8.

```
1  function decrypt( item, privacifyMessage ) {
2      var plainText = "Mensagem privada.";
3      try {
4          if( key ) {
5              var header        = getPrivacifyMessageHeader( privacifyMessage );
6              var cipherMessage = getPrivacifyCipherMessage( privacifyMessage );
7              var myCipherPwd   = header ? header[ currentUserUID ] : null;
8              var password;
9
10             if( myCipherPwd ) {
11                 password  = key.decrypt( myCipherPwd );
12                 plainText = sjcl.decrypt( password, cipherMessage );
13             }
14         }
15     } catch( error ) {
16         if( debugErrorsOn ) {
17             alert( error );
18         }
19     }
20     replaceItemContent( item, plainText );
21 }
```

Figura 3.20 – Trecho de Código para Leitura de uma Mensagem Protegida.

O trecho de código da Figura 3.20 mostra a implementação da função **`decrypt`**. Na linha 1 a função é declarada e recebe como parâmetros o elemento DOM que contém o conteúdo protegido e a *Privacify-Message*. Na linha 2, a variável **`plainText`** é inicializada com o valor *"Mensagem Privada"*. Este é o valor padrão que é exibido para o usuário caso a mensagem não seja destinada ao usuário ativo. A linha 4 verifica se a chave do usuário ativo foi inicializada e as linhas 5 e 6 obtêm o cabeçalho e a mensagem original cifrada, respectivamente, a partir da *Privacify-Message*. Na linha 7, o segredo aleatório utilizado para criptografar a mensagem original é obtido e armazenado na variável **`myCipherPwd`**, porém está criptografado com a chave pública do usuário ativo. Caso o usuário ativo não estiver relacionado no cabeçalho da mensagem, então o valor nulo é atribuído para a variável **`myCipherPwd`**. A linha 10 verifica se o segredo aleatório criptografado para o usuário ativo foi encontrado no cabeçalho da mensagem. Caso positivo, o segredo aleatório original é obtido na linha 11 e utilizado na linha 12, para obter a mensagem original e

atribuir para a variável **`plainText`**. Por fim, o conteúdo da variável **`plainText`** é utilizado para substituir o valor original do elemento HTML **`item`**, linha 20.

3.5 Considerações Finais

Este Capítulo apresentou o mecanismo para extensão da privacidade em redes sociais online. A abordagem apresentada combina criptografia com mensagens específicas para garantir que os dados do usuário sejam mantidos privados. A escolha da criptografia dos dados foi feita para que o mecanismo seja eficaz contra ataques ao servidor da rede social e de *crawling*. Os sistemas de criptografia simétrica AES e de chave pública RSA foram escolhidos por sua ampla utilização e por fornecerem um nível de segurança suficiente para validar o mecanismo proposto (AUMÜLLER *et al.*, 2002; SEIFERT, 2005; MATTHEWS, 2006; GENNARO *et al.*, 2008; NATALE *et al.*, 2008; MOZAFFARI-KERMANI; REYHANI-MASOLEH, 2010).

Adicionalmente uma implementação prova de conceito foi apresentada para a rede social Orkut com uma extensão para o navegador da *Web Google Chrome*. A implementação possibilita que mensagens no formato do *Privacify* sejam enviadas para a rede social, ao invés dos dados originais. No Capítulo 5 é apresentado um estudo de caso, que utiliza as mensagens no formato *Privacify*. Além disto, são apresentadas também as limitações e uma análise do mecanismo apresentado neste Capítulo.

Capítulo 4

RESULTADOS E ANÁLISE

Este capítulo apresenta a análise e os resultados no que diz respeito à sobrecarga de tamanho nas mensagens, proteção da privacidade dos usuários e limitações do trabalho. Um estudo de caso na forma de uma aplicação social também é apresentado.

4.1 Considerações Iniciais

No Capítulo 3 foi apresentado o mecanismo que permite o envio de mensagens criptografadas para uma rede social, com o intuito de proteger a privacidade do usuário, mantendo a possibilidade dos usuários receberem serviços personalizados. Este Capítulo apresenta uma aplicação social que oferece informações do clima onde o usuário está localizado e sugestões de presentes, de acordo com o perfil do usuário protegido pelo *Privacify*. Também são apresentados os resultados relativos ao tempo e ao tamanho no processo de geração das mensagens e, por fim, é apresentada uma análise dos resultados e das limitações do trabalho.

O Capítulo está organizado da seguinte forma: na seção 4.2 é apresentada a aplicação social e suas características; na seção 4.3 são apresentados os resultados e a análise dos resultados; a seção 4.4 apresenta as limitações do trabalho, e o Capítulo é encerrado na seção 4.5 com as considerações finais.

4.2 Estudo de Caso

O estudo de caso desta seção tem como objetivo apresentar uma aplicação social que lê alguns dados do perfil do usuário que estão protegidos no formato *Privacify* e oferece serviços personalizados. Trata-se de uma aplicação que oferece a previsão do tempo e uma lista de presentes para o usuário ativo, levando-se em consideração a localização, idade e sexo do usuário.

A Figura 4.1 ilustra a tela do aplicativo social. Ao acessar o aplicativo, os dados de localização, idade e sexo do usuário são recuperados a partir do perfil por meio da API *OpenSocial* do Orkut[22]. Em seguida, a aplicação social verifica se estes dados estão no formato *Privacify-Message*, apresentado no Capítulo 3. Caso os dados estejam neste formato, a aplicação social lê o campo *Dados Agregados* para poder oferecer os serviços.

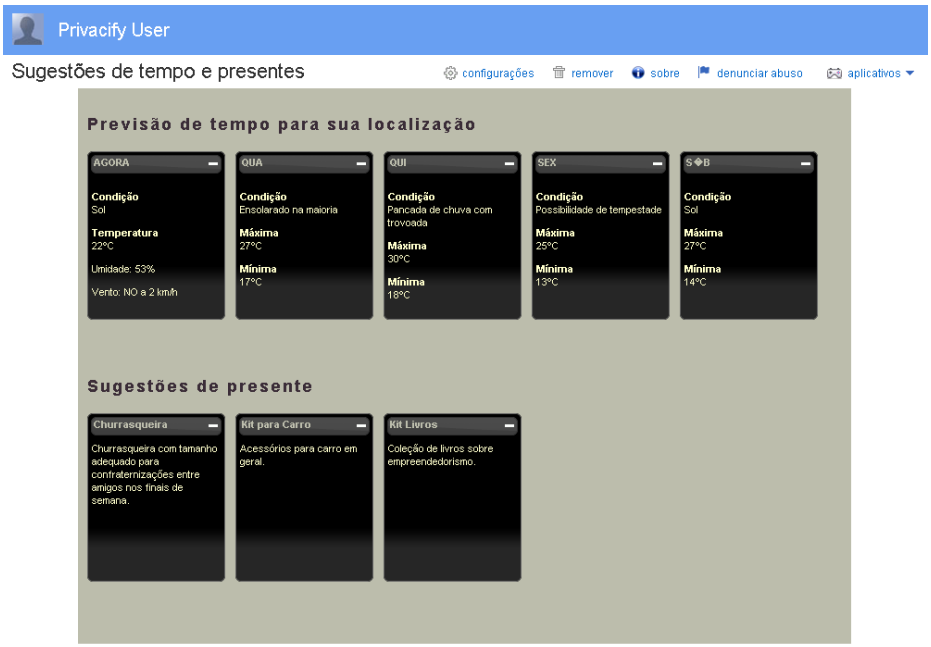


Figura 4.1 – Aplicação Social para Serviço de Clima e Sugestão de Presentes.

Na parte superior da Figura 4.1 está a previsão do tempo para a localização do usuário. No entanto, a aplicação social não tem ciência sobre o quão preciso é

[22] Disponível em: <http://code.google.com/intl/pt-BR/apis/opensocial/docs/0.8/reference/>. Acesso em: 08/06/2011.

este dado de localização, pois o nível de privacidade selecionado pelo usuário fica armazenado somente no *Servidor Privacify*. De maneira parecida, na parte inferior da Figura 4.1, uma lista de presentes é sugerida de acordo com a idade e sexo do usuário.

É importante notar que a aplicação social, assim como a rede social, não possui acesso direto ao *Privacify-Server*. Como discutido no Capitulo 3, o formato do campo *Dados Agregados* deve ser bem conhecido e neste caso definiu-se três tipos de dados para a implementação prova de conceito deste trabalho.

Um é o tipo de dado *Localização*, que representa a localização por meio de latitude e longitude. Quando o usuário está alterando os dados do perfil, a localização real é criptografada e adicionada no campo *Mensagem Cifrada* da *Privacify-Message*, e o campo *Dados Agregados* é preenchido com a latitude e longitude correspondente ajustada com uma precisão que varia de acordo com o nível de privacidade configurado. A técnica para calcular a nova localização baseia-se no trabalho de (RIBEIRO, 2009), em que um círculo, definido por um raio r e com centro na localização real, fornece uma área com os possíveis valores de localização desviados.

Os outros dois tipos de dados são *Sexo* e *Idade*. Para o tipo de dados *Idade*, definiu-se que seriam aceitos intervalo de valores. Por exemplo, para o nível de privacidade “Baixo”, um usuário com 28 anos teria a idade representada pelo intervalo “25-30”, enquanto para o nível de privacidade “Médio” teria a idade representada pelo intervalo “20-30” e “20-35” para o nível “Alto”. O tipo de dado *Sexo* traz uma exceção, pois não se trata de um tipo de dado em que a precisão possa ser diminuída. No entanto, para que a aplicação social pudesse ler e utilizar adequadamente este dado utilizou-se a idéia de valor enumerado, que aceita o caractere “M” para representar um usuário do sexo Masculino e o caractere “F” para representar um usuário do sexo Feminino.

Para a implementação da aplicação social, foi utilizado um serviço de clima do Google para oferecer as informações sobre e a lista de presentes foi implementada utilizando linguagem PHP e banco de dados MySQL.

A aplicação social apresentada nesta seção mostra que é possível, com uma pequena adaptação – a interpretação dos dados agregados da *Privacify-Message*, a utilização do *Privacify* para oferecer serviços personalizados.

A seção 4.3 apresenta alguns dados relativos ao tamanho e tempo de geração das mensagens.

4.3 Resultados

Esta seção apresenta algumas medições feitas em relação ao tamanho e ao tempo de geração das mensagens. Com isto, é possível comparar e avaliar a sobrecarga que o *Privacify* adiciona na utilização da rede social.

Os testes foram realizados em um computador desktop convencional com processador *Intel® Core™ 2 Duo*, 2.53GHz e 2 GB de memória *Random Access Memory* (RAM), que acredita-se ser uma configuração adequada para medição de tempos no que diz respeito a acesso a redes sociais online. Foram efetuados dois tipos de medições, de tempo e tamanho. As medições de tempo têm o objetivo de apresentar a sobrecarga de tempo imposta ao usuário na utilização da rede social e foram separadas em três partes: tempo de geração de chaves, tempo para cifrar as mensagens e tempo para decifrar as mensagens. As medições de tamanho têm o objetivo de apresentar a sobrecarga de armazenamento imposta ao servidor da rede social. Todos os testes foram executados 30 vezes, com objetivo de apresentar os valores médios e os desvios associados.

Tabela 4.1 – Dados de Tempos Médios para a Geração do Par de Chaves RSA.

Tamanho da Chave (bits)	Tempo Médio (segundos)	Desvio Padrão (segundos)
512	0,0704	0,01925
1024	0,8034	0,21940
2048	10,1074	4,55606

Tabela 4.1 mostra os tempos médios de geração do par de chaves pública e privada. Conforme explicado no Capítulo 3, as chaves são geradas pelo sistema no primeiro acesso do usuário, portanto não são operações que consomem tempo do usuário a todo acesso ou a cada envio de mensagem. Para chaves RSA de 512 bits e 1024 bits o tempo de geração foi abaixo de 1 segundo, o que é quase imperceptível para o usuário. No entanto, para chaves RSA de 2048 bits o tempo de

geração foi em média 10 segundos, porém, embora seja um tempo elevado, é efetuada somente no primeiro acesso do sistema.

Depois de gerada as chaves RSA, a criptografia ocorre em dois momentos durante a composição da *Privacify-Message*. Portanto, foram medidos o tempo da criptografia da mensagem original por meio do AES e a criptografia do segredo aleatório, utilizado para criptografar a mensagem original, por meio de RSA.

Tabela 4.2 – Dados de Tempos Médios para Cifrar Texto - AES.

Bloco AES (bits)	Tamanho MSG (bytes)	Tempo Médio (ms)	Desvio (ms)
128	60	15,2	1,137
128	160	15,6	1,562
128	280	15,7	1,676
192	60	15,8	1,579
192	160	15,9	1,408
192	280	16,1	1,181
256	60	15,8	1,621
256	160	16,7	1,242
256	280	16,9	1,251

A Tabela 4.2 mostra os dados de tempo médio relativos à criptografia de mensagens com o método AES para diferentes tamanhos de blocos e mensagens. Como as mensagens utilizadas para comunicação entre amigos em redes sociais são geralmente pequenas (THELWALL; WILKINSON, 2010), as medições foram realizadas a partir de mensagens com 3 tamanhos distintos com conteúdo variado similares as frases *"Feliz aniversário, paz e saúde!"*, *"Oi, tudo bem? Vamos assistir ao mais novo lançamento de filme hoje na casa da Alice."*, entre outros.

Tabela 4.3 – Dados de Tempos Médios para Cifrar Texto - RSA.

Chave (bits)	Segredo (bits)	Tempo Médio (ms)	Desvio (ms)
512	128	1,7	0,998
512	192	1,9	1,607
512	256	3,1	1,749
1024	128	4,9	2,435
1024	192	4,9	3,265
1024	256	5,1	2,778
2048	128	14,9	5,501
2048	192	15,4	5,554
2048	256	15,7	4,797

A Tabela 4.3 mostra que, nos testes realizados, a diferença de tempo médio entre a geração mais rápida e a mais lenta é inferior a 2 milésimos de segundo. Portanto, na prática, qualquer tamanho de bloco seria aceitável para a criptografia

simétrica, tendo em vista que o tempo de espera do usuário para cifrar uma mensagem é na ordem de milésimos de segundo.

Os tempos médios para cifrar os segredos aleatórios, utilizados para criptografar a mensagem original, também foram medidos e estão apresentados na Tabela 4.3. Foram definidos 3 tamanhos de segredo 128, 192 e 256 bits e combinados com os tamanhos de chaves RSA de 512, 1024 e 2048 bits. Como esperado, à medida que aumenta-se o tamanho da chave, aumenta-se também o tempo para cifrar o segredo aleatório. No entanto, a combinação mais lenta, RSA 2048 bits com segredo 256 bits, ficou na ordem de milésimos de segundos, tornando este processo aceitável, pois seria quase imperceptível para o usuário. Portanto, para enviar uma mensagem protegida, um usuário teria que esperar em média 30 milésimos de segundos no pior caso, que, novamente, parece ser aceitável levando em consideração que trata-se de uma operação bastante rápida.

Tabela 4.4 – Dados de Tempos Médios para Decifrar Texto - AES.

Bloco (bits)	Tamanho MSG (bytes)	Tempo Médio (ms)	Desvio (ms)
128	60	1,3	0,789
128	160	1,4	0,489
128	280	1,8	0,897
192	60	1,4	0,761
192	160	1,8	0,979
192	280	2,3	1,345
256	60	2,2	1,194
256	160	2,5	1,384
256	280	2,7	1,441

A medição de tempos médios para decifrar os textos é apresentada nas Tabelas 4.4 e 4.5. Ao contrário do que acontece com o processo de cifrar a mensagem, que é feito um por vez conforme a interação do usuário, o processo para decifrar as mensagens protegidas é feito automaticamente. Esta é uma característica importante, pois em uma página pode haver diversos conteúdos protegidos e decifrá-los pode ser uma tarefa que consome algum tempo.

A Tabela 4.4 mostra os valores de tempo médio para decifrar a mensagem original com o AES. Pode-se perceber que a sobrecarga de tempo imposta é menor que a sobrecarga do processo para cifrar a mensagem, sendo desta forma, também imperceptível para o usuário.

Tabela 4.5 – Dados de Tempos Médios para Decifrar Texto - RSA.

Chave (bits)	Tamanho MSG (bits)	Tempo Médio (ms)	Desvio (ms)
512	128	55,4	6,354
512	192	54,6	6,921
512	256	108,1	7,702
1024	128	383,4	11,792
1024	192	386,1	10,101
1024	256	386,2	11,892
2048	128	2.871,8	14,313
2048	192	2.875,9	36,301
2048	256	2.887,7	44,858

Ao contrário do processo para decifrar a mensagem original, o processo para decifrar o segredo aleatório adiciona uma sobrecarga de tempo maior que a sobrecarga do processo para cifrar a mensagem. A utilização de chaves de 2048 bits se torna quase proibitiva, pois uma página com 10 mensagens protegidas demoraria em média 30 segundos para ser decifrada. Neste caso, a melhor opção para segurança e desempenho seria a utilização de chaves com tamanho de 1024 bits, pois atende os níveis de segurança e tempo. É importante notar que as medições de tempos são específicas de um computador e de uma implementação das bibliotecas de criptografia, sendo que estas medições podem revelar valores diferentes em outros computadores e com outras implementações de AES e RSA.

Após a apresentação da sobrecarga de tempo imposta pelo *Privacify* ao usuário da rede social, a sobrecarga do tamanho da mensagem é analisada nas Tabelas 4.6 e 4.7.

Tabela 4.6 – Sobrecarga no Cabeçalho da Mensagem.

Chave RSA (bits)	Chave AES (bits)	Chave AES Criptografada (bits)
512	128	512
512	192	512
512	256	512
1024	128	1024
1024	192	1024
1024	256	1024
2048	128	2048
2048	192	2048
2048	256	2048

A sobrecarga de tamanho é a soma da sobrecarga do cabeçalho, da sobrecarga da mensagem cifrada e dos campos de dados agregados. Como a

sobrecarga de dados agregados varia dependendo do tipo de dado da mensagem, apresenta-se as sobrecargas de cabeçalho e de mensagem.

A Tabela 4.6 mostra os dados de sobrecarga no cabeçalho da mensagem. A chave AES é o segredo aleatório utilizado para cifrar a mensagem original. Como uma chave de n bits pode criptografar até n bits e as chaves AES utilizadas são no máximo 256 bits, o tamanho da sobrecarga é o próprio tamanho da chave RSA.

De maneira similar, o tamanho da mensagem cifrada é apresentado na Tabela 4.7. A coluna mensagem cifrada apresenta a soma do tamanho em bytes da mensagem cifrada e algumas informações adicionais para que o processo de decifrar a mensagem possa ser feito, como por exemplo, o vetor de inicialização.

Tabela 4.7 – Sobrecarga na Mensagem.

Chave AES (bits)	Mensagem Original (bytes)	Mensagem Cifrada (bytes)
128	60	145
128	160	281
128	280	450
192	60	152
192	160	288
192	280	457
256	60	152
256	160	288
256	280	457

Após apresentar os resultados das medições e efetuar uma análise sobre estes dados, as limitações do trabalho são avaliadas na seção 4.4.

4.4 Análise e Limitações

Esta seção apresenta uma análise sobre o mecanismo, identificando quais tipos de ataques podem ser mitigados e as limitações inerentes do mecanismo.

O propósito do mecanismo é estender a privacidade oferecida nas redes sociais, para que os usuários possam restringir acesso de certos conteúdos também em relação às redes sociais e terceiros, mantendo a possibilidade de oferecimento de serviços personalizados. A utilização de criptografia, juntamente com um formato de mensagens bem conhecido e uma infra-estrutura de servidor tornou possível criptografar os dados do usuário, mantendo a funcionalidade da rede social e

mantendo a possibilidade de fornecer serviços personalizados, como foi apresentado no estudo de caso na seção 4.2.

Porém, nem todos os tipos de dados do perfil do usuário são protegidos. Embora o mecanismo não imponha nenhuma restrição, a implementação feita neste trabalho não protege dados binários que o usuário envia para a rede social, como fotos e vídeos. Neste caso, técnicas de esteganografia podem contribuir para enviar uma *Privacify-Message* embutida em uma imagem, porém as técnicas empregadas devem garantir que o processamento efetuado pelas redes sociais sobre estes conteúdos, por exemplo, redimensionamento de figuras, não corrompa a *Privacify-Message*.

Outro tipo de dado que não é contemplado na proteção são as ligações estruturais. Isto significa que a lista de amigos e as interações feitas entre amigos continuam legíveis para a rede social e para terceiros, mesmo que o conteúdo da interação seja protegido. Por exemplo, caso um usuário envie uma mensagem para um de seus amigos na rede social, o conteúdo da mensagem pode ser protegido pelo *Privacify*, porém terceiros poderão verificar que esta interação foi realizada, entretanto sem poder ler a mensagem original.

Como citado no Capítulo 3, o *Cliente* e o *Servidor Privacify* são elementos seguros no mecanismo e se um destes dois elementos for comprometido, a privacidade do usuário também estará comprometida. Por exemplo, o *Cliente Privacify* ser explorado por brechas de segurança de *Javascript* (JANG *et al.*, 2010) e o *Servidor Privacify* pode ter chaves de usuários legítimos alteradas.

Em relação à manutenção das chaves públicas e privadas, escolheu-se embutir esta carga no mecanismo para torná-lo independente. Porém, pode-se adicionar o papel de uma autoridade certificadora para evitar esta sobrecarga no *Servidor Privacify*. Neste caso, o mecanismo deve adicionar uma extensão para que a comunicação entre o *Servidor Privacify* e a autoridade certificadora possa ser realizada.

Levando em consideração que os dois componentes do mecanismo são seguros, para que um adversário consiga obter acesso aos conteúdos protegidos pelo *Privacify* é necessário obter a senha utilizada para acessar a rede social e a senha para decifrar a chave privada do usuário. Como os usuários tendem a proteger suas senhas, acredita-se que a única forma de um adversário conseguir

estas senhas é pelo método de força bruta, que consiste em tentar diversas combinações de senha até obter sucesso.

Como a rede social não é um elemento seguro no mecanismo, um adversário pode conseguir acesso aos dados de diversas formas, como por exemplo, acesso não autorizado aos dados do servidor – invasão do servidor, *crawling* de parte da rede social (CHAU *et al.*, 2007), entre outras. Ao obter os dados criptografados, o adversário terá a tarefa adicional de decifrá-los e para isto, poderá decifrar o segredo aleatório criptografado com a chave pública do usuário e então utilizá-lo para decifrar o conteúdo original ou então poderá decifrar a mensagem original diretamente. Sabe-se que isto é uma tarefa que demanda tempo, mas que é possível computacionalmente (KALISK, 2003).

Desta forma, deve-se alterar as chaves dos usuários em certos períodos de tempos de acordo com o tamanho da chave escolhida. Isto pode ser uma tarefa demorada, pois muito conteúdo pode ter sido enviado para a rede social, mas trata-se de uma tarefa que pode ser feita sob demanda, conforme os conteúdos vão sendo acessados por parte dos usuários as alterações de chaves podem ser efetuadas.

Por fim, uma análise importante a ser feita é em relação à utilização do mecanismo para o mal, por exemplo, um adversário pode utilizar o mecanismo proposto para fins que sejam diferentes do propósito inicial de proteger a privacidade dos usuários. Desta forma, uma estrutura adicional é necessária para que estes dados possam ser lidos por autoridades, caso o uso do mecanismo represente uma ameaça a segurança dos usuários. Isto pode ser útil, por exemplo, em uma investigação policial em que as autoridades necessitam obter o conteúdo original de uma mensagem.

Para este fim, uma chave especial deve ser adicionada em cada mensagem protegida. Esta chave especial refere-se à autoridade que será possibilitada de ler os conteúdos protegidos e possui, neste caso, privilégio de acesso a todas as mensagens com o propósito único de investigação. Isto previne que esta chave especial seja utilizada para outros fins.

Com a inclusão de uma chave especial para possibilitar a leitura por autoridades, pode-se dizer que o mecanismo possibilita a auditoria externa. Em termos do funcionamento do mecanismo, os *Clientes Privacify* teriam o

comportamento padrão modificado nos aspectos de leitura e escrita das mensagens protegidas.

No que diz respeito à escrita de mensagens protegidas, os *Clientes Privacify* devem receber e anexar as chaves especiais das autoridades nas mensagens protegidas que são enviadas para as redes sociais. Para a leitura, uma autoridade utiliza uma senha correspondente para a utilização das chaves especiais que concedem direito a leitura das mensagens.

4.5 Considerações Finais

Este capítulo apresentou um estudo de caso na forma de uma aplicação social para o Orkut, que teve como objetivo mostrar que os usuários podem estender a privacidade das redes sociais mantendo a possibilidade de receber serviços personalizados. Também foram apresentados os resultados de medições de tempo e tamanho, que implicam em sobrecarga na geração das mensagens do *Privacify* e por fim foram apresentadas as limitações do trabalho. O próximo Capítulo apresenta os trabalhos relacionados.

Capítulo 5

TRABALHOS RELACIONADOS

Este capítulo apresenta uma visão geral dos trabalhos relacionados com a privacidade do usuário em redes sociais online.

5.1 Considerações Iniciais

Para proteger a privacidade dos usuários em redes sociais online, diversos trabalhos foram propostos nos últimos anos. Neste Capítulo são apresentados alguns trabalhos relacionados que têm como objetivo proteger a privacidade do usuário de maneira total ou parcial.

O Capítulo está organizado da seguinte forma: a seção 5.2 apresenta os trabalhos relacionados que protegem a privacidade dos usuários na utilização de aplicações sociais; na seção 5.3 são apresentados alguns trabalhos que protegem a privacidade do usuário por meio de controle de acesso; os trabalhos relacionados que utilizam criptografia para a proteção da privacidade são apresentados na seção 5.4, e por fim, a seção 5.5 apresenta as considerações finais do Capítulo.

5.2 Privacidade na Utilização de Aplicações Sociais

Esta seção apresenta os trabalhos relacionados que têm como objetivo proteger a privacidade do usuário em relação às aplicações sociais. Como discutido no Capítulo 2, as aplicações sociais fazem parte das redes sociais online e são em

sua maioria, utilitários e jogos que permitem o entreterimento dos usuários. O fato de as redes sociais não oferecerem ferramentas ao usuário para que a configuração de privacidade possa ser feita em relação às aplicações sociais, implica que uma grande quantidade de dados dos usuários como, por exemplo, dados do perfil, lista de amigos, acabam sendo reveladas para os desenvolvedores das aplicações sociais com o intuito que o serviço possa ser oferecido corretamente.

Felt e Evans (2008) notaram que grande parte das aplicações sociais poderia continuar funcionando se a rede social online fornecesse apenas uma linguagem de marcação específica e acesso aos dados do grafo social dos usuários de maneira anônima. Desta forma, a aplicação social não teria acesso a nenhum dado do usuário por meio de APIs e ao invés disto, *tags* de marcação específicas seriam utilizadas e substituídas por valores reais pela rede social no momento do acesso por parte do usuário.

Limitar todas as informações que uma aplicação social pode acessar pode melhorar a segurança dos dados do usuário, porém também pode limitar a utilidade das aplicações. O mecanismo de acesso a dados proposto por Besmer *et al.* (2009) permite que os usuários especifiquem quais atributos uma aplicação social pode acessar, além de possuir um esquema de configuração que permite especificar quais informações da lista de amigos podem ser acessadas por estas aplicações sociais. Os autores afirmam que a abordagem é eficaz para usuários que se preocupam com a privacidade, porém elementos adicionais são necessários para que o mecanismo possa proteger usuários menos atentos a questões de privacidade.

De maneira parecida, Delgado, Rodríguez e Llorente (2010) apresentam uma abordagem em que as aplicações sociais podem definir quais informações são necessárias para o seu funcionamento correto e o usuário pode conceder ou revogar o acesso a tais informações.

Os trabalhos relacionados apresentados nesta seção estabelecem que a rede social online trata-se de um elemento confiável e, portanto, é responsável por garantir a privacidade dos dados dos usuários, ao contrário da premissa assumida pelo mecanismo apresentado no Capítulo 3, que assume um servidor da rede social não é confiável, protegendo os dados do usuário por meio de criptografia implementada nos clientes com suporte de um servidor confiável.

5.3 Privacidade por meio de Controle de Acesso

Esta seção apresenta alguns trabalhos relacionados que protegem a privacidade do usuário nas redes sociais online por meio do controle de acesso.

Tootoonchian *et al.* (2008) argumentam que as informações da rede social devem ser separadas dos mecanismos de entrega, evitando que usuários mantenham várias versões inconsistentes de sua rede social. Para isto, os autores introduzem o conceito de atestados sociais e de lista de controle de acesso social (*Social Access Control List – Social* ACL). O atestado social garante a um usuário ou outra entidade um relacionamento social reconhecido pelo usuário que emitiu o atestado, por exemplo, irmão, primo, entre outros. Para completar o esquema, a lista de controle de acesso social concede as permissões de acesso aos dados sociais para os usuários que possuem o relacionamento específico com o usuário dono dos dados.

Kodeswaran e Viegas (2010) propõem uma abordagem apoiada em políticas de acesso para possibilitar a utilização dos dados agregados de usuários para fins científicos com garantias de privacidade. Com esta proposta, os usuários podem especificar quem pode ter acesso aos dados sociais e o modo de acesso. Os autores definem três modos de acesso que possibilitam a visualização completa dos dados, visualização dos dados em mais alto nível, por exemplo, cidade ou estado do usuário ao invés da localização exata e por fim visualização estatística dos dados, que trata-se de uma forma de obter os dados agregados.

Pode-se dizer que o mecanismo apresentado no Capítulo 3 controla o acesso dos usuários individualmente em cada *Privacify-Message*, referenciando os usuários no cabeçalho da mensagem. Desta forma, o acesso aos dados originais é garantido somente aos usuários autorizados.

5.4 Privacidade por meio de Criptografia

Esta seção apresenta alguns trabalhos relacionados que utilizam abordagens com criptografia para proteger a privacidade do usuário nas redes sociais online.

Lucas e Borisov (2008) descrevem uma arquitetura para mitigar os riscos de violação de privacidade em redes sociais online. A arquitetura proposta pelos autores combina criptografia simétrica e assimétrica para possibilitar a comunicação de uma-para-um e um-para-muitos entre os usuários das redes sociais online. Como prova de conceito, os autores implementaram uma aplicação para o Facebook que permite a comunicação entre os usuários.

Guha, Tang e Francis (2008) propõem uma abordagem com criptografia e divisão/agrupamento de dados privados para proteger a privacidade do usuário. O esquema funciona por meio da divisão de informações privadas em partes chamadas de "átomos" e a permuta de "átomos" entre usuários. Desta forma, usuários não autorizados visualizam pedaços de informações que não correspondem ao usuário ativo e usuários autorizados visualizam o conteúdo original. Para isto, os autores fornecem um método para mapear corretamente os "átomos" aos conteúdos originais.

Anderson *et al.* (2009) apresentam uma arquitetura em camadas para proteger a privacidade do usuário em relação a outros usuários e à própria rede social. A arquitetura proposta pelos autores é capaz de proteger dados do perfil do usuário e de relações sociais por meio de criptografia, porém requer uma re-implementação completa das redes sociais online existentes. Embora a arquitetura possa ser interessante do ponto de vista do usuário, as redes sociais online podem optar por não implementar a arquitetura, pois teriam acesso restrito aos dados do usuário.

Baden *et al.* (2009) propõem um esquema de política de acesso em que os usuários controlam o acesso aos dados e não a rede social. Os autores utilizam criptografia baseada em atributos - *Attribute-Based Encryption* (ABE) (BETHENCOURT *et al.*, 2007) para possibilitar que os dados criptografados possam ser consumidos somente por usuários que possuem as chaves com certos atributos. Com esta abordagem, para que os usuários continuem recebendo serviços personalizados é necessário conceder acesso aos dados explicitamente para que as partes envolvidas, por exemplo, as aplicações sociais.

De maneira parecida, Jahid, Mittal e Borisov (2011) propõem uma arquitetura que utiliza ABE para criptografar os dados dos usuários, porém de maneira mais eficiente para revogar acesso de um usuário, se comparado com Baden *et al.* (2009). A proposta utiliza um *proxy* que participa no processo de decifrar um

conteúdo e aplicar as políticas de acesso. Este abordagem evita a necessidade de criptografar novamente e gerar novas chaves caso o acesso de um usuário seja revogado, fato que ocorre na proposta de Baden *et al.* (2009).

5.5 Considerações Finais

Este Capítulo apresentou alguns trabalhos relacionados que utilizam abordagens diferentes para a proteção da privacidade do usuário em redes sociais online. De maneira geral, os trabalhos visam proteger a privacidade e manter a utilidade das redes sociais, possibilitando que o usuário continue recebendo serviços personalizados. Algumas propostas confiam na rede social, porém alguns trabalhos modelam a rede social como elemento não seguro. Para o segundo caso, trabalhos que modelam a rede social como elemento não seguro, a utilização de criptografia com algum complemento é geralmente utilizada e isto caracteriza o trabalho, como o mecanismo apresentado no Capítulo 3. Assim, o mecanismo apresentado no Capítulo 3 introduz uma maneira diferente, se comparado com os trabalhos relacionados, de proteger a privacidade do usuário mantendo a possibilidade de personalização.

Capítulo 6

CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho apresentou um mecanismo para estender a privacidade em redes sociais online. O mecanismo proposto utiliza a privacidade oferecida pelas redes sociais online como base e inclui uma camada de privacidade que protege dados pessoais do perfil do usuário e dados textuais de interações entre o usuário e os amigos na rede social. Esta proteção dos dados é efetiva contra terceiros, contra a rede social e contra os usuários não autorizados.

O mecanismo baseia-se em métodos de criptografia e em um formato de mensagem bem conhecido, que é manipulado por um programa no cliente para ler o conteúdo original caso este usuário possua acesso. Além disto, conta com um servidor para o armazenamento das chaves e preferências de privacidade do usuário. A rede social e terceiros podem ler o campo dados agregados da mensagem para oferecerem serviços personalizados, porém este campo contém informações que variam a precisão de acordo com o nível de privacidade preferido pelo usuário.

Como para o oferecimento de serviços personalizados necessita-se de dados pessoais do usuário e com a revelação dos dados pessoais pode não existir privacidade total, acredita-se que esta abordagem é efetiva e faz um balanço entre a revelação dos dados e a obtenção de serviços personalizados com qualidade pelo usuário, que pode ser ajustado de acordo com o nível de privacidade.

Desta forma, o objetivo inicial de estender a privacidade das redes sociais, fornecendo uma proteção adicional ao usuário em relação aos dados pessoais contra as redes sociais e terceiros, mantendo a possibilidade do oferecimento de serviços personalizados foi atingido. Adicionalmente, foi verificado na prática a utilização do mecanismo em um ambiente real de rede social online, por meio de uma implementação de prova de conceito que prepara os dados originais no formato da mensagem criptografada com os dados agregados e de um estudo de caso feito na forma de uma aplicação social para o Orkut que leva em consideração este formato de mensagem. No estudo de caso foi possível acessar os dados agregados de localização, de idade e do sexo do usuário para oferecer serviços de clima e de lista de presentes.

Embora o mecanismo proteja os dados pessoais do perfil e os conteúdos das interações entre os usuários contra os ataques de *crawling* e invasão do servidor da rede social, existe a limitação de que informações do grafo social continuam disponíveis tanto para a rede social quanto para terceiros.

Outra questão importante é a respeito da escalabilidade do mecanismo proposto. Para isto, uma análise em relação à sobrecarga de tempo e armazenamento foi efetuada, levando-se a crer que a solução é viável em termos de tempo de espera para o usuário e de armazenamento para chaves com o tamanho limitado. Para chaves RSA com tamanho igual a 2048 bits, o processo de cifrar e decifrar as mensagens são lentos, o que coloca em dúvida a utilização do mecanismo proposto. Porém, os resultados de tempo referem-se a uma implementação em particular, utilizando-se bibliotecas Javascript que possuem poder de processamento limitado para uma tarefa que exige alto processamento. Em outras implementações pode-se utilizar outros recursos para cifrar e decifrar mensagens, por exemplo, implementações em linguagens em mais baixo nível e que possuem maior eficiência para estes tipos de operações. Com isto, pode-se otimizar o tempo e torna-se viável a utilização de chaves com tamanho maiores, aumentando o intervalo de tempo em que os usuários devem trocar as chaves.

Como trabalhos futuros, pode-se avançar no sentido de proteção da privacidade de todos os dados pessoais do usuário na rede social online, por exemplo, fotos, lista de amigos, entre outros. No caso de dados binários pode-se investigar a integração do emprego de técnicas de esteganografia com o mecanismo proposto. O assunto de privacidade traz diversos desafios, pois parte dos dados

devem ser liberados ainda assim, para que os serviços personalizados possam ser oferecidos.

REFERÊNCIAS

AHN, Y.-Y. et al. Analysis of topological characteristics of huge online social networking services. In: INTERNATIONAL CONFERENCE ON WORLD WIDE WEB, 16, 2007, Banff, Alberta, Canada. **Proceedings...** ACM 2007. p. 835-844.

ANDERSON, J. et al. Privacy-enabling social networking over untrusted networks. In: WORKSHOP ON ONLINE SOCIAL NETWORKS, 2, 2009, Barcelona, Spain. **Proceedings.** ACM 2009. p. 1-6.

ASUNCION, A. U.; GOODRICH, M. T. Turning privacy leaks into floods: surreptitious discovery of social network friendships and other sensitive binary attribute vectors. In: WORKSHOP ON PRIVACY IN THE ELECTRONIC SOCIETY, 9, 2010, Chicago, Illinois, USA. **Proceedings...** ACM 2010. p. 21-30.

AUMÜLLER, C. et al. Fault Attacks on RSA with CRT: Concrete Results and Practical Countermeasures. In: INTERNATIONAL WORKSHOP ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS, 4, 2002, San Francisco Bay (Redwood City), USA. **Proceedings...** CHES 2002. p. 260-275.

BADEN, R. et al. Persona: an online social network with user-defined privacy. In: CONFERENCE ON DATA COMMUNICATION, 2009, Barcelona, Spain. **Proceedings...** ACM SIGCOMM 2009. p. 135-146.

BESMER, A.; LIPFORD, H. Tagged photos: concerns, perceptions, and protections. In: INTERNATIONAL CONFERENCE EXTENDED ABSTRACTS ON HUMAN FACTORS IN COMPUTING SYSTEMS, 27, 2009, Boston, MA, USA. **Proceedings...** ACM 2009. p. 4585-4590.

BESMER, A.; LIPFORD, H. R. Moving beyond untagging: photo privacy in a tagged world. In: INTERNATIONAL CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 28, 2010, Atlanta, Georgia, USA. **Proceedings...** ACM 2010. p. 1563-1572.

BESMER, A. et al. Social applications: exploring a more secure framework. In: SYMPOSIUM ON USABLE PRIVACY AND SECURITY, 5, 2009, Mountain View, California. **Proceedings...** ACM 2009. p. 1-10.

BETHENCOURT, J.; SAHAI, A.; WATERS, B. Ciphertext-Policy Attribute-Based Encryption. In: Proceedings of the 2007 IEEE Symposium on Security and Privacy, 2007, Washington, DC, USA. **Proceedings...** IEEE Computer Society 2007. p. 321-334.

BOYD, D. M. **Friends, Friendsters, and MySpace Top 8: Writing Community Into Being on Social Network Sites.**, 2006. Disponível em: < http://www.danah.org/papers/FriendsFriendsterTop8.pdf >. Acesso em: 17/02/2011.

BOYD, D. M.; ELLISON, N. B. **Social network sites: definition, history, and scholarship**. Journal of Computer-Mediated Communication, p. 210-230, 2008.

BROWN, G. et al. Social networks and context-aware spam. In: CONFERENCE ON COMPUTER SUPPORTED COOPERATIVE WORK, 2008, San Diego, CA, USA. **Proceedings...** ACM 2008. p. 403-412.

CHAU, D. H. et al. Parallel crawling for online social networks. In: INTERNATIONAL CONFERENCE ON WORLD WIDE WEB, 16, 2007, Banff, Alberta, Canada. **Proceedings...** ACM 2007. p. 1283-1284.

CHEN, X.; SHI, S. A literature review of privacy research on social network sites. In: INTERNATIONAL CONFERENCE ON MULTIMEDIA INFORMATION NETWORKING AND SECURITY, 2009, **Proceedings...** IEEE Computer Society 2009. p. 93-97.

COMSCORE. **Orkut continues to lead Brazil's social networking market, Facebook audience grows fivefold**. 2010. Disponível em: < http://www.comscore.com/Press_Events/Press_Releases/2010/10/Orkut_Continues_to_Lead_Brazil_s_Social_Networking_Market_Facebook_Audience_Grows_Fivefold/(language)/eng-US >. Acesso em: 05/03/2011.

CONTI, G.; SOBIESK, E. An honest man has nothing to fear: user perceptions on web-based information disclosure. In: SYMPOSIUM ON USABLE PRIVACY AND SECURITY, 3, 2007, Pittsburgh, Pennsylvania. **Proceedings...** ACM 2007. p. 112-121.

DANEZIS, G. Inferring privacy policies for social networking services. In: WORKSHOP ON SECURITY AND ARTIFICIAL INTELLIGENCE, 2, 2009, Chicago, Illinois, USA. **Proceedings...** ACM 2009. p. 5-10.

DELGADO, J.; RODRÍGUEZ, E.; LLORENTE, S. User's privacy in applications provided through social networks. In: WORKSHOP ON SOCIAL MEDIA, 2, 2010, Firenze, Italy. **Proceedings...** ACM SIGMM 2010. p. 39-44.

DWYER, C.; HILTZ, S. R.; PASSERINI, K. Trust and privacy concern within social networking sites: A comparison of Facebook and MySpace. In: CONFERENCE ON INFORMATION SYSTEMS, 13, 2007, Keystone, Colorado. **Proceedings...** AMCIS 2007. p. 1-12.

ELLISON, N. B.; LAMPE, C.; STEINFIELD, C. **FEATURE: Social network sites and society: current trends and future possibilities**. interactions**,** v. 16, n. 1, p. 6-9, 2009.

FANG, L.; LEFEVRE, K. Privacy wizards for social networking sites. In: INTERNATIONAL CONFERENCE ON WORLD WIDE WEB, 19, 2010, Raleigh, North Carolina, USA. **Proceedings...** ACM 2010. p. 351-360.

FELT, A.; EVANS, D. Privacy Protection for Social Networking Platforms. In: WEB 2.0 SECURITY AND PRIVACY, 2008, Oakland, CA, USA. **Proceedings...** W2SP 2008. p. 1-8.

FIELDING, R. T. **Architectural styles and the design of network-based software architectures**. 2000. 162 f. Tese (Doutorado em Information and Computer Science), University of California, Irvine, 2000.

GENNARO, R. et al. Threshold RSA for dynamic and ad-hoc groups. In: INTERNATIONAL CONFERENCE ON ADVANCES IN CRYPTOLOGY, 27, 2008, Istanbul, Turkey. **Proceedings...** EUROCRYPT 2008. p. 88-107.

GEORGE, A. **Living online: The end of privacy? New Scientist**, 2006. Disponível em: < http://www.newscientist.com/article/mg19125691.700-living-online-the-end-of-privacy.html >. Acesso em: 15/06/2011.

GROSS, R.; ACQUISTI, A. Information revelation and privacy in online social networks. In: WORKSHOP ON PRIVACY IN THE ELECTRONIC SOCIETY, 2005, Alexandria, VA, USA. **Proceedings...** ACM 2005. p. 71-80.

GUHA, S.; TANG, K.; FRANCIS, P. NOYB: privacy in online social networks. In: WORKSHOP ON ONLINE SOCIAL NETWORKS, 1, 2008, Seattle, WA, USA. **Proceedings...** ACM 2008. p. 49-54.

HAY, M. et al. **Anonymizing Social Networks**. SCIENCE**,** v. 245, p. 17, 2007.

HUBER, M. et al. Cheap and automated socio-technical attacks based on social networking sites. In: WORKSHOP ON ARTIFICIAL INTELLIGENCE AND SECURITY, 3, 2010, Chicago, Illinois, USA. **Proceedings...** ACM 2010. p. 61-64.

HUBER, M. et al. Exploiting social networking sites for spam. In: CONFERENCE ON COMPUTER AND COMMUNICATIONS SECURITY, 17, 2010, Chicago, Illinois, USA. **Proceedings...** ACM 2010. p. 693-695.

JAGATIC, T. N. et al. **Social phishing**. Commun. ACM**,** v. 50, n. 10, p. 94-100, 2007.

JAHID, S.; MITTAL, P.; BORISOV, N. EASiER: encryption-based access control in social networks with efficient revocation. In: SYMPOSIUM ON INFORMATION, COMPUTER AND COMMUNICATIONS SECURITY, 6, 2011, Hong Kong, China. **Proceedings...** ACM 2011. p. 411-415.

JANG, D. et al. An empirical study of privacy-violating information flows in JavaScript web applications. In: CONFERENCE ON COMPUTER AND COMMUNICATIONS SECURITY, 17, 2010, Chicago, Illinois, USA. **Proceedings...** ACM 2010. p. 270-283.

KAAFAR, M. A.; MANILS, P. Why spammers should thank Google? In: WORKSHOP ON SOCIAL NETWORKS SYSTEMS, 3, 2010, Paris, France. **Proceedings...** ACM 2010. p. 1-6.

KALISK, B. **TWIRL and RSA Key Size**. 2003. Disponível em: < http://islab.oregonstate.edu/koc/ece575/rsalabs/twirl.pdf >. Acesso em: 07/06/2011.

KODESWARAN, P.; VIEGAS, E. A policy based infrastructure for social data access with privacy guarantees. In: IEEE INTERNATIONAL SYMPOSIUM ON POLICIES FOR DISTRIBUTED SYSTEMS AND NETWORKS, 2010, Fairfax, VA. **Proceedings...** IEEE Computer Society 2010. p. 14-17.

KOROLOVA, A. et al. Link privacy in social networks. In: CONFERENCE ON INFORMATION AND KNOWLEDGE MANAGEMENT, 17, 2008, Napa Valley, California, USA. **Proceedings...** ACM 2008. p. 289-298.

KRISHNAMURTHY, B.; WILLS, C. E. Characterizing privacy in online social networks. In: WORKSHOP ON ONLINE SOCIAL NETWORKS, 1, 2008, Seattle, WA, USA. **Proceedings...** ACM 2008. p. 37-42.

LINDAMOOD, J. et al. Inferring private information using social network data. In: INTERNATIONAL CONFERENCE ON WORLD WIDE WEB, 18, 2009, Madrid, Spain. **Proceedings...** ACM 2009. p. 1145-1146.

LUCAS, M. M.; BORISOV, N. FlyByNight: mitigating the privacy risks of social networking. In: WORKSHOP ON PRIVACY IN THE ELECTRONIC SOCIETY, 7, 2008, Alexandria, Virginia, USA. **Proceedings...** ACM 2008. p. 1-8.

MATTHEWS, A. A totally self-checking s-box architecture for the advanced encryption standard. In: INTERNATIONAL SYMPOSIUM ON QUALITY ELECTRONIC DESIGN, 7, 2006, San Jose, CA. **Proceedings...** IEEE Computer Society 2006. p. 519-524.

MISLOVE, A. et al. Measurement and analysis of online social networks. In: CONFERENCE ON INTERNET MEASUREMENT, 7, 2007, San Diego, California, USA. **Proceedings...** ACM SIGCOMM 2007. p. 29-42.

MOZAFFARI-KERMANI, M.; REYHANI-MASOLEH, A. **Concurrent Structure-Independent Fault Detection Schemes for the Advanced Encryption Standard**. IEEE Trans. Comput.**,** v. 59, n. 5, p. 608-622, 2010.

NATALE, G. D. et al. A reliable architecture for the advanced encryption standard. In: EUROPEAN TEST SYMPOSIUM, 13, 2008, Verbania. **Proceedings...** IEEE Computer Society 2008. p. 13-18.

PETTERSSON, J. S. et al. How ordinary internet users can have a chance to influence privacy policies. In: CONFERENCE ON HUMAN-COMPUTER INTERACTION, 4, 2006, Oslo, Norway. **Proceedings...** ACM 2006. p. 473-476.

POLAKIS, I. et al. Using social networks to harvest email addresses. In: WORKSHOP ON PRIVACY IN THE ELECTRONIC SOCIETY, 9, 2010, Chicago, Illinois, USA. **Proceedings...** ACM 2010. p. 11-20.

POLLACH, I. **What's wrong with online privacy policies?** Commun. ACM**,** v. 50, n. 9, p. 103-108, 2007.

PUTTASWAMY, K. P. N.; SALA, A.; ZHAO, B. Y. StarClique: guaranteeing user privacy in social networks against intersection attacks. In: INTERNATIONAL CONFERENCE ON EMERGING NETWORKING EXPERIMENTS AND TECHNOLOGIES, 5, 2009, Rome, Italy. **Proceedings...** ACM 2009. p. 157-168.

RIBEIRO, F. N. **Sistema para a oferta de Serviços Baseados em Localização com Garantias de Privacidade ao Usuário**. 2009. 123 f. Dissertação (Mestrado em Sistemas Distribuídos e Redes), Universidade Federal de São Carlos, São Carlos, 2009.

RICHARDSON, L.; RUBY, S. **Restful web services**. O'Reilly 2007. 446

RIVEST, R. L.; SHAMIR, A.; ADLEMAN, L. **A method for obtaining digital signatures and public-key cryptosystems**. Commun. ACM**,** v. 21, n. 2, p. 120-126, 1978.

SCHRAMMEL, J.; KÖFFEL, C.; TSCHELIGI, M. How much do you tell?: information disclosure behaviour indifferent types of online communities. In: INTERNATIONAL CONFERENCE ON COMMUNITIES AND TECHNOLOGIES, 4, 2009, University Park, PA, USA. **Proceedings...** ACM 2009. p. 275-284.

SEIFERT, J.-P. On authenticated computing and RSA-based authentication. In: CONFERENCE ON COMPUTER AND COMMUNICATIONS SECURITY, 12, 2005, Alexandria, VA, USA. **Proceedings...** ACM 2005. p. 122-127.

SHAKIMOV, A. et al. Privacy, cost, and availability tradeoffs in decentralized OSNs. In: WORKSHOP ON ONLINE SOCIAL NETWORKS, 2, 2009, Barcelona, Spain. **Proceedings...** ACM 2009. p. 13-18.

SOLOVE, D. J. **I've got nothing to hide and other misunderstandings of privacy**. San Diego Law Review**,** v. 44, p. 745-772, 2007. Disponível em.: < http://ssrn.com/paper=998565 >. Acesso em: 10/11/2010.

THELWALL, M.; WILKINSON, D. **Public dialogs in social network sites: What is their purpose?** J. Am. Soc. Inf. Sci. Technol.**,** v. 61, n. 2, p. 392-404, 2010.

TOCH, E. et al. Locaccino: a privacy-centric location sharing application. In: INTERNATIONAL CONFERENCE ADJUNCT PAPERS ON UBIQUITOUS COMPUTING, 12, 2010, Copenhagen, Denmark. **Proceedings...** ACM 2010. p. 381-382.

TOCH, E.; SADEH, N. M.; HONG, J. **Generating default privacy policies for online social networks**. Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems. Atlanta, Georgia, USA: ACM 2010.

TOOTOONCHIAN, A. et al. Lockr: social access control for web 2.0. In: WORKSHOP ON ONLINE SOCIAL NETWORKS, 1, 2008, Seattle, WA, USA. **Proceedings...** ACM 2008. p. 43-48.

WILLIAMS, J. Social networking applications in health care: threats to the privacy and security of health information. In: WORKSHOP ON SOFTWARE ENGINEERING IN HEALTH CARE, 2010, Cape Town, South Africa. **Proceedings...** ICSE 2010. p. 39-49.

WILSON, C. et al. User interactions in social networks and their implications. In: EUROPEAN CONFERENCE ON COMPUTER SYSTEMS, 4, 2009, Nuremberg, Germany. **Proceedings...** ACM 2009. p. 205-218.

YOUNG, A. L.; QUAN-HAASE, A. Information revelation and internet privacy concerns on social network sites: a case study of Facebook. In: INTERNATIONAL CONFERENCE ON COMMUNITIES AND TECHNOLOGIES, 4, 2009, University Park, PA, USA. **Proceedings...** ACM 2009. p. 265-274.

ZHELEVA, E.; GETOOR, L. To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. In: INTERNATIONAL CONFERENCE ON WORLD WIDE WEB, 18, 2009, Madrid, Spain. **Proceedings...** ACM 2009. p. 531-540.