UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

GUILMOUR HENRIQUE DIAS ROSSI

LIBREFLIX: PLATAFORMA DE VÍDEO SOB DEMANDA PEER-TO-PEER COM RECOMENDAÇÕES QUE PRESERVAM A PRIVACIDADE PARA UM STREAMING LIVRE

CURITIBA

2025

GUILMOUR HENRIQUE DIAS ROSSI

LIBREFLIX: PLATAFORMA DE VÍDEO SOB DEMANDA PEER-TO-PEER COM RECOMENDAÇÕES QUE PRESERVAM A PRIVACIDADE PARA UM STREAMING LIVRE

Libreflix: A Peer-to-Peer On-demand Video Platform with Privacy-Preserving Recommendations for a Free Streaming

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Sistemas de Informação do Curso de Bacharelado em Sistemas de Informação da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Luiz Celso Gomes Jr.

CURITIBA

2025

GUILMOUR HENRIQUE DIAS ROSSI

LIBREFLIX: PLATAFORMA DE VÍDEO SOB DEMANDA PEER-TO-PEER COM RECOMENDAÇÕES QUE PRESERVAM A PRIVACIDADE PARA UM STREAMING LIVRE

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Sistemas de Informação do Curso de Bacharelado em Sistemas de Informação da Universidade Tecnológica Federal do Paraná.

Data de aprovação: 27/maio/2025

Prof. Dr. Daniel Fernando Pigatto
Doutorado
Universidade Tecnológica Federal do Paraná

Profª. Drª. Leyza Elmeri Baldo Dorini
Doutorado
Universidade Tecnológica Federal do Paraná

Prof. Dr. Luiz Celso Gomes Junior
Doutorado
Universidade Tecnológica Federal do Paraná

CURITIBA
2025

à mamãe,
que deu seu tudo e
assim nunca me faltou nada.

AGRADECIMENTOS

Primeiramente, expresso minha sincera gratidão à minha família, minha mãe, Laudilena Dias da Silva, e meu pai, Claudemir Rossi, pelo amor incondicional, dedicação e sacrifício, que são a força motriz por trás de cada conquista.

Agradeço à minha eterna companheira, Karolyna Gutierres, pelo apoio ao longo de tantos anos e, principalmente, por todas as conversas, mundanas e transcendentais, que foram alívio e alimento para rumar em direção a tudo aquilo que acredito.

Agradeço ao meu orientador e professor Luiz Celso Gomes Jr., que foi quem primeiro me mostrou a magia da computação e a realização de uma vida acadêmica em busca dos saberes. Lembro-me de um aluno franzino, inseguro com a universidade; chega um novo professor aplicado e solícito e esse aluno começa a enxergar a graduação com outros olhos - amando computação e conquistando sua primeira grande nota. Para mim, isso mudou tudo...

Agradeço ao Laboratório de Tecnologias Livres (LabLivre) da Universidade Federal do ABC (UFABC), que, por meio do Edital 01/2018 de fomento a softwares livres culturais, possibilitou o desenvolvimento de várias funcionalidades do projeto Libreflix. Agradeço à banca de avaliação daquele edital: André Filipe de Assunção e Brito, Bruna Moreira, Ludimila Bela Cruz, Murilo Bansi Machado, Yorik van Havre; agradecendo também à coordenação e inspiração do professor Sérgio Amadeu da Silveira.

Por último, mas não menos importante, agradeço a todas as pessoas que colaboraram com o projeto Libreflix de alguma maneira: pessoas programadoras, designers, arquivistas, cinéfilas e doadoras que participaram das campanhas de financiamento coletivo[1]. Nominalmente, agradeço também a Marcielo Lopes De Morais, Julio Lira, Matheus Cavalcante de Oliveira, Gabriel Polastrini, Mateus "N2OMatt", Augusto Silva e Ayrton "ayr-ton".

É um prazer criar com vocês. Meu imenso obrigado.

[1] Uma lista nunca-completa de agradecimentos do projeto está disponível em: https://libreflix.org/agradecimentos

Dentro del capitalismo no hay solución para la vida; fuera del capitalismo hay incertidumbre, pero todo es posibilidad. Nada puede ser peor que la certeza de la extinción. Es momento de inventar, es momento de ser libre, es momento de vivir bien.

Ana Esther Ceceña
(CECEÑA, 2012)

RESUMO

Este trabalho propõe-se a atuar em duas frentes complementares. A primeira, que constitui o núcleo principal desta pesquisa, apresenta o Libreflix, uma plataforma de *streaming* de vídeo sob demanda, voltada para a publicação e o consumo gratuito de conteúdos audiovisuais. Diferentemente das plataformas tradicionais, geralmente controladas por corporações que impõem políticas restritivas de direitos autorais e cobram pelo acesso, o Libreflix agrega conteúdos com licenças permissivas, promovendo a colaboração, o engajamento comunitário e a democratização do conhecimento na internet. Descreve-se a arquitetura da plataforma, suas principais funcionalidades e são discutidos aspectos relevantes e lições aprendidas ao longo do seu desenvolvimento contínuo. A segunda frente deste trabalho explora o uso de Ambientes de Execução Confiáveis (Trusted Execution Environments - TEE), com ênfase na tecnologia Intel SGX, para a gestão de sistemas de recomendação em Redes Sociais Online, e nesse caso, no próprio Libreflix. Esta abordagem busca proteger os dados dos usuários e prevenir seu uso indevido, sem comprometer a funcionalidade ou a sustentabilidade financeira dessas redes. São apresentadas a arquitetura da solução proposta e análises de desempenho que orientam a escolha de algoritmos de recomendação compatíveis com a execução segura em Ambientes de Execução Confiáveis.

Palavras-chave: streaming livre; libreflix; sistemas de recomendação; distribuição de vídeo peer-to-peer; cultura digital.

ABSTRACT

This work aims to address two complementary fronts. The first, which constitutes the core of this research, presents Libreflix, an on-demand video streaming platform designed for the free publication and consumption of audiovisual content. Unlike traditional platforms, typically controlled by corporations that impose restrictive copyright policies and charge for access, Libreflix aggregates content under permissive licenses, fostering collaboration, community engagement, and the democratization of knowledge on the internet. The platform's architecture and its main functionalities are described, along with relevant aspects and lessons learned throughout its ongoing development. The second front of this work explores the use of Trusted Execution Environments (TEE), with an emphasis on Intel SGX technology, for the management of recommendation systems in Online Social Networks - in this case, within Libreflix itself. This approach seeks to protect user data and prevent its misuse, without compromising the functionality or financial sustainability of these networks. The architecture of the proposed solution and performance analyses are presented, providing guidance for selecting recommendation algorithms suitable for secure execution within Trusted Execution Environments.

Keywords: free streaming; libreflix; recommender systems; peer-to-peer video distribution; digital culture.

LISTA DE ABREVIATURAS E SIGLAS

Abreviaturas

art.	Artigo
cap.	Capítulo
sec.	Seção

Siglas

OSN	Redes Sociais Online, do inglês *Online Social Networks*
P2P	par-a-par, do inglês *peer-to-peer*
SGX	do inglês *Software Guard Extensions*
TEE	Ambientes de Execução Confiáveis, do inglês *Trusted Execution Environments*
UTFPR	Universidade Tecnológica Federal do Paraná

SUMÁRIO

1 INTRODUÇÃO

As plataformas de vídeo sob demanda e as redes sociais online tornaram-se elementos centrais na forma como interagimos, consumimos e produzimos conteúdos na internet. Ambas as tecnologias moldam profundamente as experiências culturais e comunicacionais contemporâneas, mas também impõem desafios importantes relacionados à liberdade de acesso, privacidade e uso ético dos dados. Este trabalho busca abordar essas questões em duas frentes complementares: de um lado, a democratização do acesso à cultura audiovisual por meio de um streaming livre (ROSSI, 2019); de outro, a proteção dos dados dos usuários frente aos riscos associados à exploração indevida de informações pessoais em sistemas de recomendação.

1.1 *Libreflix: A Peer-to-Peer On-demand Video Platform for Free Streaming*

Na primeira parte deste trabalho (ROSSI; GOMES-JR, 2019), consideramos em como as plataformas de vídeo sob demanda tornaram-se um dos meios mais populares para distribuir e acessar conteúdos multimídia. Um relatório da Cisco (CISCO, 2018) estima que, em 2022, o streaming de vídeo foi responsável por 82% de todo o tráfego da internet. O conteúdo oferecido nessas plataformas típicas é, em sua maioria, composto por obras com todos os direitos reservados pelos detentores de direitos autorais. Esse conteúdo é gerenciado por empresas de mídia, como Netflix e Amazon, que cobram uma taxa para que os usuários possam acessar suas coleções. As barreiras relacionadas aos direitos autorais e aos custos financeiros podem ser excessivamente restritivas, dificultando a participação de criadores e consumidores nesse desenvolvimento cultural. Seguindo os passos do movimento do Software Livre e os objetivos do Creative Commons, é importante garantir o acesso livre (no sentido de liberdade) às obras multimídia, tanto para consumidores quanto para criadores, sem infringir as leis de direitos autorais. O acesso livre à cultura digital é um passo fundamental para a democratização do conhecimento.

Para isso, o objetivo do primeiro trabalho é apresentar, discutir e validar a plataforma Libreflix para vídeo sob demanda. O Libreflix foi criado em 2017 com a finalidade de catalogar obras multimídia que podem ser transmitidas livremente pela internet (ou seja, sem restrições de direitos autorais). A plataforma visa fomentar a cultura multimídia nas comunidades online, tornando-se um ponto de referência para o consumo e a publicação de conteúdos livres. O projeto implementou diversas funcionalidades para alcançar seus objetivos, como a transmissão de conteúdo via redes peer-to-peer (P2P), múltiplas ferramentas de descoberta de conteúdo orientadas ao usuário e instrumentos para o engajamento comunitário. O trabalho também discute questões importantes relacionadas ao financiamento (da plataforma e dos criadores), ao licenciamento do código-fonte, às lições aprendidas e às perspectivas futuras.

1.2 *Privacy-preserving recommendations for Online Social Networks using Trusted Execution Environments*

Na segunda parte deste trabalho, (ROSSI *et al.*, 2018) discutimos como tornaram-se onipresentes em nosso cenário tecnológico, seja por meio de serviços dedicados (por exemplo, Facebook, LinkedIn), seja por meio de funcionalidades sociais incorporadas a sites de diversos setores da indústria e do governo (por exemplo, seções de comentários, links compartilháveis). As interações possibilitadas pelas OSNs são claramente benéficas para os usuários, proporcionando entretenimento, comunicação, autoexpressão (de opiniões, interesses, serviços profissionais etc.) e descoberta de produtos e serviços (com base nas preferências de usuários com interesses semelhantes). Entretanto, ao utilizar esses serviços, os usuários deixam um vasto rastro de dados que pode ser acessado e utilizado em práticas antiéticas (FIRE; GOLDSCHMIDT; ELOVICI, 2014). Exemplos recentes dessas práticas podem ser encontrados em eleições e referendos de grandes democracias, onde agentes mal-intencionados direcionaram campanhas a cidadãos para disseminar desinformação e promover interesses políticos relacionados aos resultados das urnas (ESPANHA *et al.*, 2024).

Uma forma de evitar esses problemas seria criptografar todos os dados dos usuários nessas aplicações. Contudo, essa solução é impraticável, pois o modelo de negócios da maioria das OSNs baseia-se no cruzamento das preferências dos usuários com anúncios publicitários. Assim, uma solução para esses problemas deve combinar maior privacidade para os usuários, sem deixar de oferecer oportunidades de receita para os provedores de serviço. Neste trabalho, descrevemos uma arquitetura para OSNs que utiliza Ambientes de Execução Confiáveis (Trusted Execution Environments - TEEs) com o objetivo de proteger os dados dos usuários contra acessos não autorizados.

Em nossa solução, nem mesmo os provedores de serviço ou os anunciantes podem acessar diretamente os dados dos usuários. Em vez disso, oferecemos recomendações de conteúdo e publicidade como serviços em nossa Interface de Programação de Aplicações (API). Nesse cenário, os provedores de serviço ainda podem recomendar conteúdo aos usuários (por exemplo, filmes na Libreflix ou postagens no Facebook), e os anunciantes podem adequar seus anúncios às preferências dos usuários. As implementações atuais de TEEs possuem restrições relacionadas ao aumento de sobrecarga e à memória principal limitada disponível para o processo. Portanto, é importante avaliar o comportamento dos TEEs no cenário proposto. Neste trabalho, apresentamos testes preliminares de desempenho de diversos algoritmos de recomendação executados no Intel SGX. O objetivo é determinar os algoritmos que sofrem menor degradação sob SGX e avaliar sua aplicabilidade na plataforma proposta.

1.3 Publicação dos Artigos

Ambos os trabalhos foram revisados por pares em revisão cega, aceitos e apresentados em dois simpósios nacionais. O trabalho do Apêndice A foi publicado nos Anais Estendidos do XXV Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia) (ROSSI; GOMES-JR, 2019). O trabalho do Apêndice B foi publicado nos Anais do XVIII Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg) (ROSSI *et al.*, 2018).

REFERÊNCIAS

CECEÑA, A. E. Dominar la naturaleza o vivir bien: disyuntiva sistémica. **Debates urgentes**, v. 1, n. 1, p. 117–129, 2012.

CISCO, V. Cisco visual networking index: Forecast and trends, 2017–2022. **White Paper**, v. 1, 2018.

ESPANHA, M. B. *et al.* O escândalo cambridge analytica: Influência dos algoritmos que violam a privacidade nas redes sociais. **REVISTA LATINO-AMERICANA DE GESTÃO, TECNOLOGIA E SOCIEDADE**, v. 1, n. 1, 2024.

FIRE, M.; GOLDSCHMIDT, R.; ELOVICI, Y. Online social networks: Threats and solutions. **IEEE Communications Surveys and Tutorials**, v. 16, n. 4, p. 2019–2036, 2014.

ROSSI, G. Free streaming: Thinking on a digital distribution of the commons in the streaming era. *In*: **Anais do I Congresso Internacional sobre o Comum e os Commons**. [*S.l.*: *s.n.*], 2019.

ROSSI, G.; GOMES-JR, L. Libreflix: A peer-to-peer on-demand video platform for free streaming. *In*: **Anais Estendidos do XXV Simpósio Brasileiro de Sistemas Multimídia e Web**. Porto Alegre, RS, Brasil: SBC, 2019. p. 137–141. ISSN 2596-1683. Disponível em: https://sol.sbc.org.br/index.php/webmedia_estendido/article/view/8152.

ROSSI, G. *et al.* Privacy-preserving recommendations for online social networks using trusted execution environments. *In*: **Anais do XVIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais**. Porto Alegre, RS, Brasil: SBC, 2018. p. 41–48. Disponível em: https://sol.sbc.org.br/index.php/sbseg/article/view/4268.

APÊNDICE A – Libreflix: A Peer-to-Peer On-demand Video Platform for Free Streaming

Libreflix: A Peer-to-Peer On-demand Video Platform for Free Streaming

Guilmour Rossi
Universidade Tecnológica Federal do Paraná (UTFPR)
Curitiba, Brasil
guilmour@alunos.utfpr.edu.br

Luiz Gomes-Jr
Universidade Tecnológica Federal do Paraná (UTFPR)
Curitiba, Brasil
gomesjr@dainf.ct.utfpr.edu.br

ABSTRACT

On-demand video streaming platforms are becoming one of the most important means to publish and consume multimedia content on the internet. The platforms are usually owned by corporations that charge for content access and implement restrictive copyright policies. While the platforms are beneficial, in many aspects, to the multimedia industry and consumers, they can be too restrictive for independent content creators and users with limited resources. This paper presents Libreflix, an on-demand video platform for free publication and consumption of content. Libreflix is intended for aggregation of content with permissive licences, fostering collaboration, community engagement and democratization of knowledge in the internet. This paper describes Libreflix's architecture, the main functionalities implemented in the platform, and discusses important aspects and lessons learned during its ongoing development.

KEYWORDS

streaming platform, free streaming, digital culture, peer-to-peer distribution

1 INTRODUCTION

On-demand video platforms have become one of the most popular means to distribute and access multimedia content. A Cisco report [2] estimates that in 2022 video streaming will account for 82% of all internet traffic. The content offered in typical platforms is, for the most part, comprised of works with all rights reserved by the copyright holder. This content is managed by media companies such as Netflix and Amazon that charge a fee for users to access their collections.

The copyright and financial barriers can be too restrictive for creators and consumers to participate in this cultural development. Following the footsteps of the Free Software movement and the Creative Commons goals, it is important to guarantee free (as in Freedom) access to multimedia works for both consumers and creators without the infringement of copyright laws. Free access to the digital culture is an important step towards the democratization of knowledge.

The goal of this paper is to present, discuss and validate the Libreflix platform for on-demand video. Libreflix was created in 2017 to catalog multimedia works that can be freely transmitted over the internet (i.e. without copyright restrictions). The platform aims at fostering multimedia culture in online communities, becoming a reference point for free content consumption and free content publication. The Librefix project has deployed several features to address its goals, such as peer-to-peer (P2P) content transmission (Section 3), several user-centered content discovery functionalities and tools for community engagement (Section 4). The paper also discusses important questions related to financing (of the platform and creators), source-code licensing, lessons learned and perspectives (Section 5).

In: XVIII Workshop de Ferramentas e Aplicações (WFA 2019), Rio de Janeiro, Brasil. Anais Estendidos do Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia). Porto Alegre: Sociedade Brasileira de Computação, 2019.
ISSN 2596-1683

2 RELATED WORK

2.1 On-demand video

On-demand video services offer catalogs of multimedia content that can be accessed at any time, providing a flexible and convenient user experience [7]. The capitalization model varies, with monthly subscriptions and pay-per-view being the most common schemes.

A key aspect underlying these services is content delivery, usually handled by data streaming protocols [6]. The protocols allow users to start consuming the content as soon as the first packages arrive, avoiding long downloads and waiting times.

2.2 Free Streaming

The Free Streaming model [10] advocates that the freedom in multimedia streaming distribution is enabled by tools (software programs, platforms, digital archives, etc.) built following the free software [12] guidelines; furthermore, the digital content itself must be amenable to be freely transmitted on the internet without violating the work's copyright. Access must also be free of charge or any other barriers.

The Creative Commons initiative[1] offers permissive licensing for multimedia content, such as share with Attribution (CC BY), which allows distribution, remixing, and adaptation of content as long as the original creator is properly credited. When a content intended for free distribution is published in a proprietary platform (such as YouTube) that prevents users from downloading the content, the permissiveness of the license is only partially fulfilled.

In this context, the Free Streaming movement emphasizes that comprehensive freedom cannot be accomplished by permissive licensing alone – the concept of freedom must also be ingrained between the mean of access and the transmitted content.

2.3 Peer-to-peer distribution

Currently, most of the streaming on the internet is based on the client-server model, where a client connects to the server to retrieve the media content. The Peer-to-Peer (P2P) model that gained

[1] https://creativecommons.org

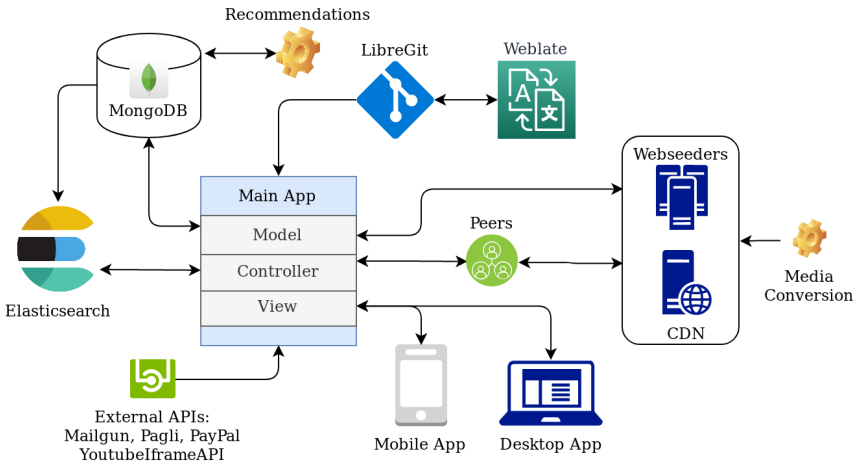


Figure 1: Libreflix architecture

traction in the 2000s, in contrast, considers users (peers) as part of the distribution infrastructure. In a P2P network any peer can serve the content previously downloaded to any other peer. This approach makes the distribution process more reliable and decentralized, reducing the load and importance of dedicated content servers.

WebTorrent [1] is an open source JavaScript library that offers P2P functionalities based on the popular BitTorrent protocol. WebTorrent uses the Web Real-Time Communication (WebRTC) suite of protocols, a W3C standard for client-to-client communication running over web-browsers (with no need for external programs) [9].

3 LIBREFLIX

Libreflix is an online platform for on-demand video created with the goal of providing free access to films of independent production. The platform allows users to browse and discover works of fiction, documentaries, and series to be watched on a variety of devices (browser, desktop, or mobile). Creators, directors and producers can use the platform for publishing and dissemination of their cultural pieces. The submission process for video content is collaborative, and starts with the creator filling a form on the website. Content management and curation is done collaboratively with the help of moderators with access to the administrative interface.

3.1 Architecture

Figure 1 shows the interactions among elements of the architecture of the Libreflix service. The website was developed in Node.js using the Model-View-Controller (MVC) design pattern. The users access the services through their devices and interact with several front-end technologies (bottom of the figure). Technologies such as Bootstrap and the template engine Nunjucks improve the user experience, providing visually appealing interfaces that are responsible to different screen sizes.

A mobile application is being developed to complement the web-based service. The implementation uses the Kotlin [8] language, which is a modern alternative for application development for Android (Figure 2).

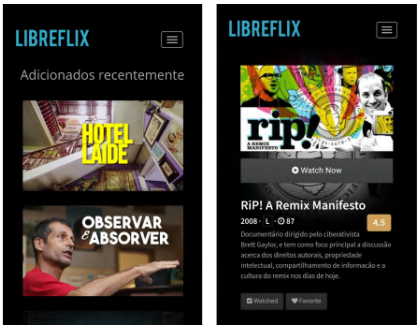


Figure 2: Screens of the mobile app

In the back-end, video processing is handled by a series of scripts using the powerful *ffmpeg* [2] tool while P2P distribution is delegated to a set of customizations based on WebTorrent. Data storage and management is performed by a MongoDB instance which implements a NoSQL hierarchical data model. Elasticsearch [4], which offers a powerful and scallable search engine, was used to provide keyword-based queries for users to retrieve titles.

Following its commitment with free access, the platform uses the free Git versioning service LibreGit[3]. The Libreflix platform and its microservices are hosted on a virtual private server (VPS) running Debian GNU/Linux.

3.2 Video Streaming

The main goal of a video streaming platform is to provide a convenient watching experience for any title, at any time, and on any device. In Libreflix, the Watch interface houses the media player for content display. Initially, the standard media player was implemented to interface with the YouTube's Player API. This approach

[2] https://ffmpeg.org [3] https://libregit.org

Figure 3: Documentary film playing within the platform

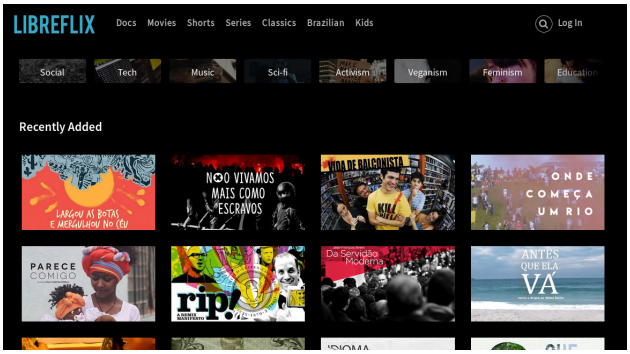


Figure 4: Platform home screen

allows the use of YouTube's transmission resources and was important at the beginning when Libreflix did not have means to handle video delivery. Another advantage is the wide offer of titles that can be included in the catalog (since YouTube is a popular platform for creators).

Libreflix is currently on an effort to offer other streaming options. A new model recently incorporated in the platform is a P2P protocol based on WebTorrent. Since both WebTorrent and Libreflix employ the Javascript programming language, it was easy to integrate the projects.

In the WebTorrent, data (video) chunks are transmitted to the client. The library renders the content directly in the browser's DOM (Document Object Model) for the video page. One issue with this approach (also common to other P2P implementations) is that the content must be shared by other users at the time of playback, requiring these users to be seeding the content or watching the same film.

A later development in the BitTorrent technology, called WebSeed, allows new peers to share the content through standard HTTP/FTP protocols [5]. These peers can then distribute data when there are not enough regular peers sharing the content. This approach has been integrated in WebTorrent as well, and was implemented in the Libreflix project.

This allows Libreflix to offer a hybrid data transmission strategy that employs seed servers whenever the number of regular peers is insufficient for high data throughput. Seed servers can be common web servers that support the network as peers and can be raised by the community related to the project, by volunteers, or other organizations interested in strength the project.

4 USER AND COMMUNITY FEATURES

This section discusses the main features offered in Libreflix with implementation details and the lessons learned in their development and deployment.

4.1 Catalog browsing

Libreflix's home page offers an exploratory interface for users to browse the available films (Figure 4). The titles are organized by several criteria such as publication time, popularity, format, duration, category or tags. The menu offers links to the most accessed categories such as documentaries, fiction, short films, series, classics, and family/kids content.

By hovering the mouse over a film's cover, the user is presented with more information about the title, such as name, director, parental rating, and duration. Clicking on the cover opens a modal window with more information on the synopsis and attributed tags for the title.

Categorization of items is challenging for any digital library. In Libreflix the categorization is centered on user convenience, in a two-level hierarchy. The top level contains the main film formats: fiction, documentaries and series. While this division does not correlate with title type or duration, it is is more likely to match user's general intent in a watching session. From the top-level categories the users access sub-categories based on the format chosen. Examples of sub-categories are teen movies (fiction), educational (documentaries), and adventure (series).

Other options for navigation were explored trough the creation of specific URL slugs. The first one allows direct access to all titles available with a certain duration length limit. This feature can be used to show films according to the available time to watch of an user. The second slug refers to a page that will show all works from a specified country, allowing the direct access to other countries audiovisual culture.

Libreflix offers a search-based interface where users can use keywords to retrieve related titles. Submitted queries can include the name of the production, direction, category, format or key words at the sinopse.

Libreflix also offers a title recommendation engine, which suggests titles based on users' previous interactions. The recommendation engine has been implemented following studies on privacy-preserving recommendation models [11]. Currently the recommendations use a collaborative filtering approach that will improve as users rate and interact with more titles.

4.2 Title interface and community engagement

When a user chooses a title she is redirected to a page containing detailed information about the work (Figure 5). This page also concentrates several community interaction features to foster user interaction, engagement, debate and networking. The main features are:

Figure 5: Page containing information about the work and enabling user interaction

- Comments field: for users to express their opinion on the title, write a synopsis, etc.
- Ratings: for users to assign a personal rating, using the 1 to 5 stars scheme often adopted in the cinema world. This data is used to calculate a global rating for each title by averaging all of its ratings.
- Favorite and Watched tags: users can use these two tags for personal organization. This data is also used by the backend in the recommender system. When a user rates a film, the film is automatically tagged as watched.
- External references: this feature allows external sources that cited the title to be displayed alongside the details for the work. External references can provide access to related synopses, news articles, reference sources, etc. The feature strengthen network connections, providing users with direct access to other perspectives and to the entire blogsphere.
- External links: this feature is intended to offer further institutional information about the title, such as official website, social network profiles, Wikipedia page, IMDB, etc. The links are an invitation for users to get to know and connect with the production.

4.3 Administrative interface

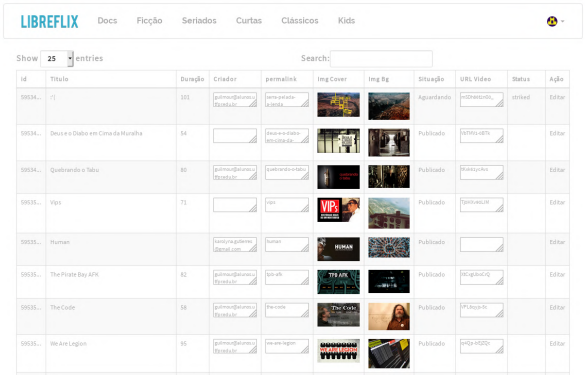


Figure 6: Administrative area for curation

Libreflix uses a collaborative approach to catalog curation. The administrative interface (Figure 6) handles most of the collaborative workflow, encompassing title submission, approval, metadata reviews, and publication. The main roles in the administrative interface are:

(1) Users submitting titles: usually the creator of the work (or someone that know the permissive licensing of it). The user fills a form with basic information about the work. In the submission page the user can follow the status of the review process and edit the metadata before the title is approved or rejected.
(2) Moderators: Moderator users have access to an interface with all titles submitted to the platform (Figure 6). The titles are organized in a spreadsheet containing key information such as work title, cover image, duration, posted by, etc. The moderator can contact users to obtain further information about their submissions. All moderators have permission to accept or reject submissions following Libreflix's guidelines. Any status change for a title is reported in an automatic email sent to the respective user.

4.4 Internationalization

A key factor for cultural interchange is the internationalization of a software or platform. According to [3], the internationalization process (i18n) is associated with the implementation of software resources to support translation and localization (l10n). Translation refers to the literal conversion of text from one language to another. Localization refers to the adaptation of content to meet regional expectations when the content is displayed.

Libreflix supports internationalization in the main components of the user interface. The Node.js library i18n is used to handle automatic translation while browser cookies store user's language preferences.

To allow users with no technical background to help in the translation process, Libreflix uses the Weblate[4] project, which allows translations to be created, discussed, compared and approved in a web-based interface. Weblate supports exporting the translations to JSON which, combined with Libreflix Git-based and open source code development, allows for streamlined publication of the translated sections.

5 FINANCE, LICENSING AND PERSPECTIVES

This section discusses important aspects of Libreflix's financing strategy, licensing model and contextualizes the current and future social impact of the platform.

5.1 Financing model

Libreflix advocates a free access model to all content served. To support the service Libreflix accepts voluntary donations from users using the platform Acredito.me[5] which allows the publication of personalized crowdfunding campaigns without retaining fees other than payment gateways.

It is also important to implement mechanisms to help in generating funding for the creators. Libreflix allows, for each registered title, a monetary retribution option to be displayed in the details

[4] https://weblate.org [5] https://acredito.me

page. When a user wants to make a donation for the production of a title, a pop-up window is shown where she can set the value of the contribution (Figure 7). The user is then redirected to services like PayPal and Pagli to continue the transaction.

Figure 7: Pop-up window to contribute with the production

5.2 Licensing model

The license in which Libreflix is distributed is the GNU Affero General Public License (AGPLv3) which makes explicit the need for access to source code even when there is no direct access to the application's executable binary (including web applications). Mobile and desktop applications also follow licenses in the free software philosophy.

The choice of license permeates all aspects of Libreflix, fostering Free Streaming and more independent content distribution. This model provides the possibility of a community process of planning, development, and learning. It also allows users to create their instances and adapt them their needs, ensuring that their improvements will be public for others. The source code for the platform can be accessed on https://libregit.org/libreflix.

5.3 Perspectives

The feedback on the Libreflix platform has been very positive. Even though registration is not mandatory, the number of registered users has surpassed 30,000. The most recent statistics (first half of 2019) shows average visits counts of around 85,000 per month.

The catalog has over 310 available works, where 75% are non-fiction works and 62% correspond to works produced in Brazil. 40% are feature films, 57% shorts and 3% are series. The moderators are constantly working to correctly process, approve, and manage the incoming submissions. The number of pending works is approximately 250.

The project's goal now is to reach more people, fostering free streaming and the dissemination of digital culture. Next steps include the development of a Smart TV application and enhancements on the mobile app.

Another interesting project is the creation of an offline version, which will allow the platform to run on community networks where internet access are unavailable or limited. Alongside that, we hope to gather more volunteers to help on moderation, translation of the software and subtitle creation.

6 CONCLUSION

This paper presented Libreflix, and on-demand, free and collaborative video streaming platform. Libreflix has evolved to become a service that integrates several tools and strategies to foster the dissemination of the digital multimedia culture. The platform is becoming a reference point for independent creators as well as consumers interested in alternative content.

By using decentralized technologies, such as P2P data sharing, the platform offers an alternative distribution paradigm giving users an appealing option beyond traditional providers and illegal (pirated) content.

Another important concern in the development of Libreflix is to integrate means for direct and voluntary support to creators. In a streamlined fashion, users can choose to contribute with small monetary donations directly to content producers.

The main instance of the Libreflix platform can be accessed in the address: https://libreflix.org. Demo and tutorial videos are available on: https://vdn.libreflix.org/demo/demo.html.

ACKNOWLEDGMENTS

This work has been supported with funding from the Laboratory of Free Technologies (LabLivre) through the Research Development Foundation (Fundep) – Grant/Edital 01/2018 LabLivre UFABC. We would also like to thank the users that voluntarily helped in Libreflix's crowdfunding campaign.

REFERENCES

[1] Feross Aboukhadijeh. 2014. WebTorrent.
[2] VNI Cisco. 2018. Cisco visual networking index: Forecast and trends, 2017–2022. *White Paper* 1 (2018).
[3] Joao Dias e Vitor Gonçalves. 2017. Localização de Software e Páginas Web. *adolesCiência* 4, 1 (2017), 85–90. https://www.adolesciencia.ipb.pt/index.php/adolesciencia/article/view/197
[4] Elastic. 2015. Hardware: Elasticsearch: The Definitive Guide. https://www.elastic.co/guide/en/elasticsearch/guide/current/hardware.html
[5] John Hoffman and DeHackEd. 2008. HTTP Seeding – BitTorrent Enhancement Proposal Nº 17. http://www.bittorrent.org/beps/bep_0017.html
[6] James F. Kurose and Keith W. Ross. 2006. *Redes de computadores e a Internet: uma nova abordagem. 3ª edição.*
[7] João Carlos Massarolo and Dario Mesquita. 2017. Video on demand: a new television platform.
[8] Kassiano Resende. 2018. *Kotlin com Android: Crie aplicativos de maneira fácil e divertida.* Casa do Código.
[9] Roberto Oliveira Rocha. 2014. WebRTC - Evolução na Web.
[10] Guilmour Rossi. 2019. Free Streaming: Thinking on a Digital Distribution of the Commons in the Streaming Era. In *Anais do I Congresso Internacional sobre o Comum e os Commons.* to appear.
[11] Guilmour Rossi, Luiz Gomes-Jr, Marcelo Rosa, and Keiko Fonseca. 2018. Privacy-preserving recommendations for Online Social Networks using Trusted Execution Environments. In *Anais do XVIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais.* SBC, Porto Alegre, RS, Brasil, 41–48. https://sol.sbc.org.br/index.php/sbseg/article/view/4268
[12] Richard Stallman. 2002. *Free software, free society: Selected essays of Richard M. Stallman.* GNU Press. https://www.gnu.org/philosophy/fsfs/rms-essays.pdf

APÊNDICE B – Privacy-preserving recommendations for Online Social Networks using Trusted Execution Environments

Privacy-preserving recommendations for Online Social Networks using Trusted Execution Environments

Guilmour Rossi, Luiz Gomes-Jr[1], Marcelo Rosa[1], Keiko Fonseca [1]

me@guilmour.org, gomesjr@dainf.ct.utfpr.edu.br, {mrosa,keiko}@utfpr.edu.br

[1]Universidade Tecnológica Federal do Paraná
Curitiba – Brazil

Abstract. *Online Social Networks (OSN) have changed how individuals interact with each other and with organizations, offering means of communication, publication and consumption of information. As OSNs have become a substantial part of users' online activities, OSN providers have understood the value of the data being generated and exploited it to maximize profits. Recently, malicious agents have invested in the manipulation of OSN data to attain commercial advantages or influence public opinion with dangerous consequences. This paper describes our ongoing efforts towards the use of Trusted Execution Environments (TEE), more specifically Intel's SGX, for the management of recommendation engines for OSNs. Our solution focuses on protection of user data and prevention of misuse without compromising OSNs' functionality nor OSNs' revenue from advertisements. We describe the architecture of our system and report performance results that can be used to guide the selection of recommendation algorithms for execution under SGX.*

1. Introduction

Online Social Networks (OSN) have become commonplace in our technological landscape, being on dedicated services (e.g., Facebook, Linkedin) or in social features included in websites from most branches of industry and government (e.g., comments sections, shareable links). The interactions enabled by OSNs are clearly beneficial for users, providing entertainment, communication, self-expression (of opinions, interests, professional services, etc.), and discovery of products and services (based on the preferences of similar users).

However, by using these services, users leave behind a large trail of data that may be accessed and used in unethical practices [4]. Recent examples of such practices can be found in elections and referendums of major democracies, where malicious agents targeted citizens to spread misinformation and advance political interests related to ballot outcome. One way to prevent these problems is to encrypt all user data in such applications. This solution is, however, impractical because the business model of most OSNs is based on matching user preferences to advertisements. A solution to these problems must then combine increased privacy for users while still offering revenue opportunities for service providers.

In this paper, we describe an architecture for OSNs that uses Trusted Execution Environments, or TEEs, in order to shield user data from unauthorized access. In our solution, not even the service providers and advertisers can access the user data directly.

Instead, we offer content and advertisement recommendations as services in our Application Program Interface (API). In this scenario, service providers can still recommend content to users (e.g., movies for Netflix or posts for Facebook), and advertisers can match their ads to users' preferences.

The current implementations of TEEs have restrictions related to increased overhead and limited main memory available to the process. Therefore, it is important to assess the behavior of TEEs in the proposed scenario. In this paper, we present preliminary performance tests of various recommendation algorithms running on Intel SGX. The goal is to determine the algorithms that degrade the least under SGX and assess their applicability in the proposed platform.

The remainder of this paper is structured as follows: Section 2 describes related work and introduces important concepts. Section 3 provides details about the proposed architecture focusing on the aspects related to social recommendations. Section 4 shows experimental results for various recommendation algorithms running inside SGX enclaves. Finally, section 5 concludes the paper.

2. Background and related efforts

Governments around the world have grown concerned with the current state of privacy in OSNs. There has been many reports of loose privacy policies allowing misuse of user data with criminal or political purposes. The European Union has been the first to pass far-reaching legislation in an attempt to curb the abusive use of personal data. The General Data Protection Regulation (GDPR)[1] aims at protecting the privacy of individual users. While those are essential legal tools to guarantee privacy, our approach focuses on the technological aspects and aims at, by design, restricting access to user data.

On the technological front, there have been many efforts to improve privacy in OSNs. Many efforts focus on anonymization of user data [1], preserving a degree of privacy when publishing the social network data. These proposals aim at preventing abuse from third-party players but offer no protection against internal attacks or data breaches.

Other proposals focus on applying blockchains to store data and allow users to manage access rights to their private information. Zyskind et al. [10] developed a protocol that relies on a distributed blockchain to store and manage user data. While granting control of the data to the users, this type of approach still hands user data to providers, opening opportunities for abuse and misuse.

More recently, advances in hardware have allowed stronger privacy guarantees. Intel's SGX platform, for example, offers programmers with a secure environment where all computation is executed over encrypted memory. These environments, called enclaves, offer privacy even in worst-case scenarios where attackers have direct access to the provider's hardware. SGX [3] is currently the most comprehensive and commercially available implementation of a trusted execution environment (TEE). As a TEE, it has abilities of (i) containing a hardware encoded cryptographic key (known only by the module); (ii) remote attestation (which means two enclaves can trust each and exchange cryptographic keys), (iii) sealing data (which means once a data is sealed by the module, it can only

[1] https://ec.europa.eu/commission/priorities/justice-and-fundamental-rights/data-protection/2018-reform-eu-data-protection-rules_en

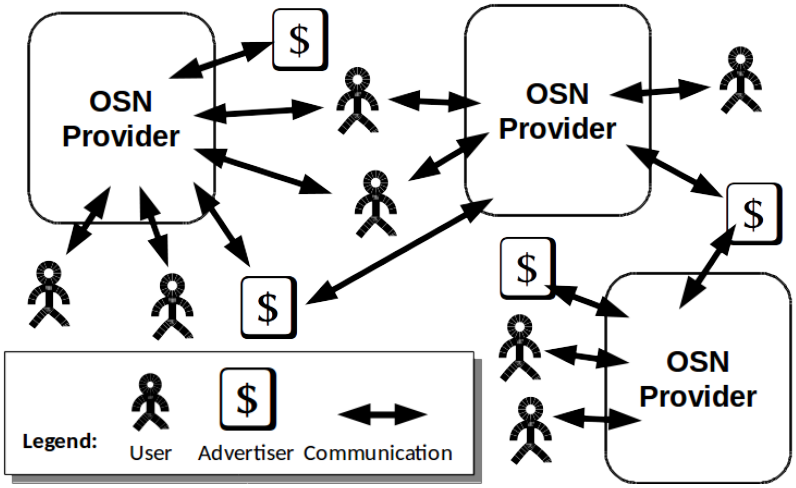


Figure 1. Overview of the interactions among the elements of the platform.

be unsealed by that module). To the developer's perspective, it ensures that nothing can tamper with the code running inside such modules.

There have been several proposals to harness these privacy guarantees for data processing platforms. OPAQUE [9] is a Spark based solution that employs SGX and obfuscation algorithms to guarantee privacy in a distributed processing context. The SecureCloud [8] project goes beyond the data processing scope offering a complete stack of tools that can be used to build SGX-based secure applications. The project offers application building blocks including programming language interpreters, communication protocols, and data management and processing services.

Neither OPAQUE nor SecureCloud intend to provide user-centered guarantees as those proposed here. The projects focus on protecting user data from attackers that may obtain physical access to provider's hardware. In their setting, there is no restriction to what providers or eventual partners can do with the data. Therefore, to properly tackle the current concerns regarding misuse of users data in OSN, it is necessary to design a more restrictive framework. In our proposal, even service providers do not have access to the social network data. To compensate for the added restrictions, the framework offers APIs for providers and advertisers to insert content into their OSNs. The framework itself is responsible for applying recommendation algorithms to suggest content and advertisement to users without the intervention of the provider.

3. Platform architecture

Our proposed architecture is based on three main roles: OSN providers, users and advertisers. OSN providers are organizations that offer services based on social network interactions. Examples of OSN providers include social networks such as Facebook and Twitter, but also encompass government and commerce websites that include social data for commentaries or recommendations. Users are persons with internet-enabled devices (smartphones, tablets, computers) that access the services of the OSN providers. Finally, advertisers are organizations that pay OSN providers to match their advertisement based on user preferences.

Figure 1 shows an example with interactions among these roles. As shown in the

figure, users can access multiple OSN providers and, likewise, advertisers can promote content in multiple OSN providers. Details about internal elements are presented in the Figure 2. The main focus of this paper is the Social Security Platform, which is the component that processes user data and is described next. We also briefly describe the functionalities of other components in the next section.

3.1. Secure Social Platform (SSP)

The Secure Social Platform (SSP) is meant to be a generic platform that covers most application cases for online social networks, such as user management and content recommendation. Social application providers will be able to download and install the platform on their own servers and manage its performance as necessary. Being based on Trusted Execution Environments (TEE) technology, the platform is effectively a black box from the point of view of the providers. Providers will be able to configure the platform to suit their services, setting up content schemas and advertisement partners, and will be able to access anonymous statistics of visualization of content and ads. All the processing of data happens inside encrypted enclaves, protecting their privacy. The functionalities for user and content administration are exposed to providers as a service API.

Providers will also be able to build the frontend of their applications with any technology they want. However, the frontend server must provide XSLT (eXtensible Stylesheet Language Transformations) templates that will be used inside the Request Broker for user's data integration. Therefore, even though the provider is free to implement the user interface however it wants, as long as it provides adequate templates to be securelly merged with user data before delivery. At no point in this process the provider has access to user data.

Users and advertises will communicate with the Request Broker through an extension of the HTTP protocol that will encompass encryption, authentication and attestation procedures. The extension is called Social HTTPS (SHTTPS).

3.2. Privacy-preserving recommendations

The focus of this paper is in exploring one critical aspect of the Secure Social Platform: the recommendation of content to users. Recommendation algorithms are used for personalization, timeline construction, friendship suggestion, and advertisement placement. This type of algorithm is the most compute-intensive task in a OSN scenario and is central to effective user interaction and advertisement revenue.

To protect user data, we employ Intel's SGX architecture, currently the most advanced Trusted Execution Environments (TEEs) technology. SGX provides secure enclaves where software code runs in an encrypted region of the computer memory that is protected from attacks related to physical memory access. SGX, being a recent development, has limitations in terms of performance degradation and limited memory allocation. Therefore, it is important to test the limits of the technology in our intended scenario.

To simplify the deployment of our algorithms in SGX-enabled environments, we used SCONE [2]. SCONE is a secure container mechanism for Docker[2] that uses the SGX trusted execution to protect container processes from outside attacks. We implemented the

[2]https://www.docker.com/

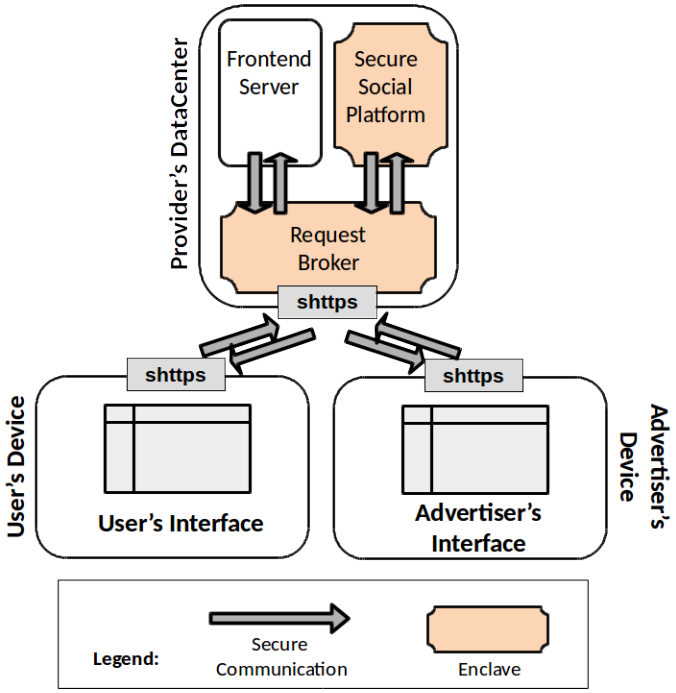


Figure 2. General architecture of the proposed platform.

service API in Python and employed recommendation algorithms from the Surprise [6] project.

4. Experimental procedure and results

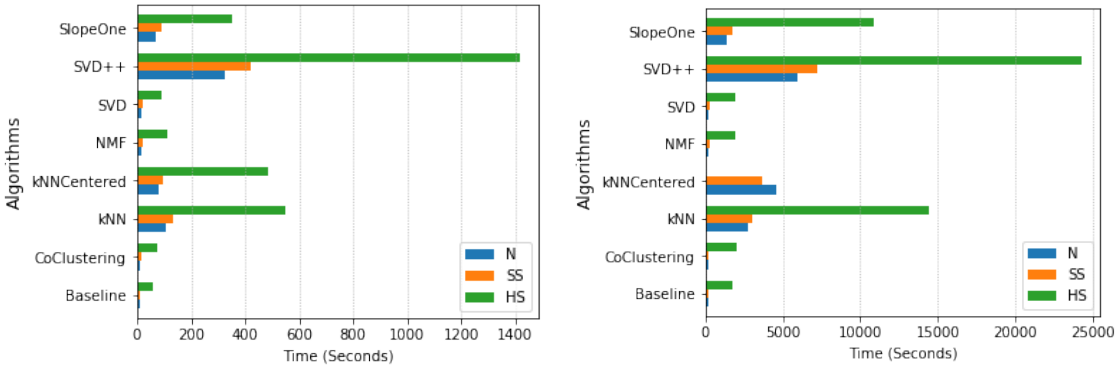
We study the performance of recommender algorithms in three distinct scenarios and using two different sizes of datasets. The three scenarios are meant to capture the performance overhead imposed by the use of SXG. The scenarios are:

1. Regular (N): running the code natively, using the standard Python interpreter.
2. Simulated SCONE (SS): interpreting the code inside the SCONE container, with the mode parameter set to simulated, where the execution proceeds with the SCONE functionalities but does not use the SGX hardware.
3. Hardware-based SCONE (HS): compilation is made in hardware mode, where the execution is forced by SCONE to run inside of the SGX enclave.

To build the prediction models, we used algorithms implemented in the Python Scikit-Surprise package [6], which offers a range of recommender algorithms, including variations of Singular-Value Decomposition (SVD) and k-Nearest Neighbor (kNN) approaches.

The two datasets used were obtained from the MovieLens datasets [5], one containing 100,000 ratings (1-5) from 943 users on 1682 movies and the other containing 1,000,209 ratings from 6,040 users on 3706 movies. We evaluated the time of basic execution of the API to train the data and build the predictions (in this case, the top 5 recommendations) for each user inside the database, including the time required by SCONE, when using it, to up the enclave and allocate memory.

Due to the limited size of the enclave page cache (EPC) [7], in this case 128 MB, we have to increase the SCONE's heap environment variable to 2 GB on the first dataset



(a) 100K Ratings dataset **(b) 1M Ratings dataset**

Figure 3. Execution time of the algorithms on three distinct scenarios. Regular (N), with simulated SCONE (SS), and with hardware-enabled SCONE (HS).

and 16 GB on the second dataset. Is out of the scope of this paper considerations about accuracy and precision of the algorithms. Our goal is to examine performance degradation related to the use of SGX and its memory limitations.

All the experiments were performed on a GNU/Linux (kernel 4.15) based computer using an Intel Xeon E3-1280 v6 CPU with 4 cores at 3.90 GHz and 8 hyper-threads counting with 8 MB cache. The computer has 32 GB of RAM and a hard disk of 2 TB. The SCONE image used the operational system Alpine[3] version 3.6 and Python 3.5.4.

Table 1. Running times of the algorithms and proportional increase.

Algorithm	Time (s)					
	100K Dataset			1M Dataset		
	N	SS	HS	N	SS	HS
Baseline	8.3	12.8	59.6 (7.2*N; 4.7*SS)	142.2	180.4	1780 (12.6*N; 9.9*SS)
CoClustering	9.3	14.4	73.6 (7.9*N; 5.2*SS)	145.6	197.4	2000.7 (13.8*N; 10.2*SS)
kNN	107.8	135.1	548.9 (5.1*N; 4.1*SS)	2751.7	3025.8	14401.3 (5.3*N; 4.8*SS)
kNNCentered	81.1	98.5	484.3 (6*N; 5*SS)	4587.3	3619.0	82921,5 (18,1*N; 23*SS)
NMF	14.1	21.8	111.5 (7.9*N; 5.2*SS)	178.7	255.9	1925.3 (10.8*N; 7.6*SS)
SVD	14.7	22.3	92.5 (6.3*N; 4.2*SS)	194.6	263.2	1966.7 (10.2*N; 7.5*SS)
SVD++	324.0	422.6	1415.1 (4.4*N; 3.4*SS)	5966.8	7213.3	24277 (18*N; 3.4*SS)
SlopeOne	69.7	92.8	353.1 (5.1*N; 3.9*SS)	1354.1	1716.5	10877 (8.1*N; 6.4*SS)

Figure 3 shows execution time results for each algorithm. Baseline is a simple algorithm based on averages of user and item ratings and will not be discussed further. From the graphs, it can be seen that running the algorithms over SGX has a clear performance penalty. Also, kNN-based algorithms in general require a longer time to execute. The SVD-based algorithms (including NMF) performed well, except for SDV++ that requires a more complex data structure to represent ratings as well as more demanding calculations to derive the predictions. In the 1M tests, the kNNCentered algorithm finish after over 23 hours.

Table 1 shows the precise running times for the algorithms and also compares the proportional increase in time for the executions inside the SGX enclave (column HS, in

[3]https://alpinelinux.org/about

Table 2. Average RMSE and MAE of the algorithms.

Algorithm	Dataset			
	100K		1M	
	RMSE	MAE	RMSE	MAE
Baseline	0.944	0.748	0.909	0.719
CoClustering	0.963	0.753	0.915	0.717
kNN	0.980	0.774	0.923	0.727
kNNCentered	0.951	0.749	0.929	0.738
NMF	0.963	0.758	0.916	0.724
SVD	0.934	0.737	0.873	0.686
SVD++	0.920	0.722	0.862	0.673
SlopeOne	0.946	0.743	0.907	0.715

Source: Scikit-Surprise documentation [6].

parenthesis). Here it can be seen a clear distinction between the algorithms based on SVD and kNN: the time for SVD algorithms degrades more (when compared with non-SGX executions) as the dataset grows. For example, comparing with regular (N) execution, SVD took 6.3 times longer to execute in the 100K and 10.2 times longer in the 1M dataset. The KNN algorithm maintained the approximately 5 times penalty throughout the tests. The CoClustering algorithm was the fastest in both datasets but, similarly to SVD methods, it degrades more when using SGX. This shows the importance of benchmarking the algorithms in the SGX context, since standard benchmarks would not reveal this SGX-specific trend.

As a reference, we included in Table 2 recommendation quality assessments for each algorithm. SVD-based algorithms tend to provide more accurate predictions in terms of Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). Therefore, this must be considered when choosing the algorithm for a task with high accuracy requirements.

5. Conclusion

This paper described our architecture for privacy-preserving Online Social Networks (OSNs), detailing the usage scenario, roles and components. The main focus of the paper is testing a critical component of OSNs: the recommendation algorithms used in several tasks such as content suggestion and ad placement. Since our platform takes advantage of Intel's SGX technology, it is important to assess the performance of recommendation algorithms in the secure environment.

Our experiments have shown that different classes of algorithms respond differently to the SGX environment. This is likely caused by SGX's enclave page cache (EPC) limitations. The tests suggest that SVD-based algorithms tent to degrade more, in relative terms, under SGX for larger datasets. However, in absolute terms, the pure SVD algorithm (and also the related NMF algorithm) was shown to be faster than kNN methods even in the larger dataset. The tests are inconclusive in whether the stronger degradation under SGX would favor kNN methods for even larger datasets (millions of users and items). To perform this type of test, we will be implementing a parallel version of the architecture and employing parallel variations of the algorithms.

Other future efforts will focus on refining the API to be applied in real world OSNs: including authentication steps, error handling, and secure secondary memory storage.

The tests presented here show the practicability of our approach. We consider the performance penalty for the best performing algorithms to be reasonable given the improved security guarantees under SGX. We expect that this type of approach will play an important role in protecting user data in OSNs.

6. Acknowledgments

This research is being performed in the context of the SecureCloud project. The SecureCloud project has received funding from the European Union's Horizon 2020 research and innovation program and was supported by the Swiss State Secretariat for Education, Research and Innovation (SERI) under grant agreement number 690111. This work was partially funded by the EU-BR SecureCloud project (MCTI/RNP 3rd Coordinated Call).

Referências

[1] J. H. Abawajy, M. I. H. Ninggal, and T. Herawan. Privacy preserving social network data publication. *IEEE Communications Surveys and Tutorials*, 18(3):1974–1997, 2016.

[2] S. Arnautov, B. Trach, F. Gregor, T. Knauth, A. Martin, C. Priebe, J. Lind, D. Muthukumaran, D. O'Keeffe, M. Stillwell, D. Goltzsche, D. M. Eyers, R. Kapitza, P. R. Pietzuch, and C. Fetzer. Scone: Secure linux containers with intel sgx. In *12th USENIX Symposium on Operating Systems Design and Implementation*, 2016.

[3] V. Costan and S. Devadas. Intel sgx explained. Technical Report 2016/086, Cryptology ePrint Archive, 2016.

[4] M. Fire, R. Goldschmidt, and Y. Elovici. Online social networks: Threats and solutions. *IEEE Communications Surveys and Tutorials*, 16(4):2019–2036, 2014.

[5] F. M. Harper and J. A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4):19:1–19:19, Dec. 2015.

[6] N. Hug. Surprise, a Python library for recommender systems. `http://surpriselib.com`, 2017.

[7] I. R. Intel. Software guard extensions sdk for linux* os, revision 1.5.

[8] F. Kelbert, F. Gregor, R. Pires, S. Köpsell, M. Pasin, A. Havet, V. Schiavoni, P. Felber, C. Fetzer, and P. R. Pietzuch. Securecloud: Secure big data processing in untrusted clouds. In D. Atienza and G. D. Natale, editors, *DATE*, pages 282–285. IEEE, 2017.

[9] W. Zheng, A. Dave, J. G. Beekman, R. A. Popa, J. E. Gonzalez, and I. Stoica. Opaque: An oblivious and encrypted distributed analytics platform. In *14th USENIX Symposium on Networked Systems Design and Implementation*, pages 283–298, 2017.

[10] G. Zyskind, O. Nathan, and A. Pentland. Decentralizing privacy: Using blockchain to protect personal data. In *IEEE Symposium on Security and Privacy Workshops*, pages 180–184. IEEE Computer Society, 2015.