Ministry of education and science of the Russian Federation
Federal state-founded educational institute of high professional education

«NATIONAL RESEARCH TOMSK POLYTECHNIC UNIVERSITY»

Institute Cybernetics
Educational programme Computer Science and Engineering
Department Software Engineering

MASTER THESIS

Research title
On Security of Mobile Messengers

UDC 004.773.6

Student

Group	Name	Signature	Date
8ВМ5И	Markus Müller		

Supervisor

Position	Name	Academic degree	Signature	Date
Associate professor	S.V. Axyonov	PhD		

CONSULTANTS:

On «Financial management, resource efficiency and resource saving» chapter

Position	Name	Academic degree	Signature	Date
Associate professor	N.O. Chistyakova	PhD		

On «Social responsibility» chapter

Position	Name	Academic degree	Signature	Date
Associate professor	Y.V. Anischenko	PhD		

PERMIT TO DEFENCE:

Head of department	Name	Academic degree	Signature	Date
Head of department	M.A. Ivanov	PhD		

Tomsk – 2017

Планируемые результаты обучения по ООП

Код Результата	**Результат обучения (выпускник должен быть готов)**
Профессиональные компетенции	
Р1	Применять глубокие естественнонаучные и математические знания для решения научных и инженерных задач в области информатики и вычислительной техники.
Р2	Применять глубокие специальные знания в области информатики и вычислительной техники для решения междисциплинарных инженерных задач.
Р3	Ставить и решать инновационные задачи инженерного анализа, связанные с созданием аппаратных и программных средств информационных и автоматизированных систем, с использованием аналитических методов и сложных моделей.
Р4	Выполнять инновационные инженерные проекты по разработке аппаратных и программных средств автоматизированных систем различного назначения с использованием современных методов проектирования, систем автоматизированного проектирования, передового опыта разработки конкурентно способных изделий.
Р5	Планировать и проводить теоретические и экспериментальные исследования в области проектирования аппаратных и программных средств автоматизированных систем с использованием новейших достижений науки и техники, передового отечественного и зарубежного опыта. Критически оценивать полученные данные и делать выводы.
Р6	Осуществлять авторское сопровождение процессов проектирования, внедрения и эксплуатации аппаратных и программных средств автоматизированных систем различного назначения.
Универсальные компетенции	
Р7	Использовать глубокие знания по проектному менеджменту для ведения инновационной инженерной деятельности с учетом юридических аспектов защиты интеллектуальной собственности.
Р8	Осуществлять коммуникации в профессиональной среде и в обществе в целом, активно владеть иностранным языком, разрабатывать документацию, презентовать и защищать результаты инновационной инженерной деятельности, в том числе на иностранном языке.
Р9	Эффективно работать индивидуально и в качестве члена и руководителя группы, в том числе междисциплинарной и международной, при решении инновационных инженерных задач.
Р10	Демонстрировать личную ответственность и ответственность за работу возглавляемого коллектива, приверженность и готовность следовать профессиональной этике и нормам ведения инновационной инженерной деятельности. Демонстрировать глубокие знания правовых, социальных, экологических и культурных аспектов инновационной инженерной деятельности.
Р11	Демонстрировать способность к самостоятельному обучению, непрерывному самосовершенствованию в инженерной деятельности, способность к педагогической деятельности.

Ministry of education and science of the Russian Federation
Federal state-founded educational institute of high professional education

«NATIONAL RESEARCH TOMSK POLYTECHNIC UNIVERSITY»

Institute Cybernetics
Educational programme Computer Science and Engineering
Department Software Engineering

APPROVED BY:
Head of department
_____ _______ M.A.Ivanov
(Signature) (Date) (Name)

TASK
for the final qualifying research

Form:

Master thesis

To student:

Group	**Name**
8ВМ5И	Markus Müller

Title:

On Security of Mobile Messengers	
Approved by the rector’s order (date, ID)	

Date of research completion:	02.06.2017

TECHNICAL TASK:

Initial data *(Product requirements, User services, Description of solution, Market analysis, Impact on the Environment)*	The aim of this thesis is to define a list of requirements for secure mobile messengers. These requirements should be presented in a protection profile conforming to ISO/IEC 15408 (Common Criteria). In addition to this a security target for Telegram should to be created. This security target should then be used for evaluating the Telegram application software to show that the requirements are applicable to existing software.
List of tasks must be presented in the thesis *(Review. Related research, Task description, Research procedure, Development and design procedures, Results obtained, Additional chapters, Appendix, Conclusion).*	Overview over technologies and methods; Related Work; Requirements for Secure Mobile Messengers; Security Target for Telegram; Technical Evaluation of Telegram; Financial management, resource efficiency and resource saving; Social Responsibility; Summary and Conclusion
List of graphical data:	Presentation

Consultants

Part	**Consultants**
Financial management, resource efficiency and resource saving	N.O. Chistyakova, Associate professor, PhD
Social responsibility	Y.V. Anischenko, Associate professor, PhD

Date of task obtaining	22.03.2017

The task was given by:

Position	Name	Academic degree	Signature	Date
Associate professor	S.V. Axyonov	PhD		

The student gets the task:

Group	Name	Signature	Date
8ВМ5И	Markus Müller		

Ministry of education and science of the Russian Federation
Federal state-founded educational institute of high professional education

«NATIONAL RESEARCH TOMSK POLYTECHNIC UNIVERSITY»

Institute Cybernetics
Educational programme Computer Science and Engineering
Educational level Master
Department Software Engineering
Research period Summer term 2016-2017

Form:

Master thesis

CALENDAR RATING PLAN
of the final qualifying research

Date of research completion:	02.06.2017

Checkpoint date	Research section	Max score
24.03.2017	*Overview technologies and related work*	*10*
14.04.2017	*Defining requirements for secure messengers*	*20*
28.04.2017	*Analyzing Telegram and defining security target*	*20*
15.05.2017	*Evaluation of Telegram messenger*	*20*
02.06.2017	*Financial management, resource efficiency and resource saving*	*10*
02.06.2017	*Social responsibility*	*10*
08.06.2017	*Presentation*	*10*

The task was given by:

Position	Name	Academic degree	Signature	Date
Associate professor	S.V. Axyonov	PhD		

ARGEED BY:

Head of department	Name	Academic degree	Signature	Date
Head of department	M.A. Ivanov	PhD		

TASK FOR CHAPTER
«FINANCIAL MANAGEMENT, RESOURCE EFFICIENCY AND RESOURCE SAVING»

To student:

Group	Name
8BM5И	Markus Müller

Institute	**Cybernetics**	**Department**	**Software Engineering**
Educational level	**Master**	**Educational programme**	**Computer Science and Engineering**

Initial data to «Financial management, resource efficiency and resource saving » chapter:	
1. *Costs of research, including technical, financial, energy, information and human costs*	*Work with related research presented in articles, journals, bulletins, and official documents*
2. *Norms of expenditure of resources*	
3. *The taxation system used, the rates of taxes, discounting and lending*	
List of tasks:	
1. *Evaluation of commercial and innovative potential*	*Analysis of potential consumers. Assessment of the quality and prospective of the project. Research planning.*
2. *Development of the charter of the technical project*	
3. *Planning of management process: structure and schedule, budget, and risks*	
4. *Estimation of resource, financial and economical efficiency*	
List of graphical data:	
1. *Assessment of the competitiveness of solution* 2. *Gantt diagram* 3. *SWOT analyses* 4. *PEST analyses*	

Date of task obtaining	

The task was given by the consultants:

Position	Name	Academic degree	Signature	Date
Associate professor	N.O. Chistyakova	PhD		

The task was accepted by the student:

Group	Name	Signature	Date
8BM5И	Markus Müller		

TASK FOR CHAPTER
«SOCIAL RESPONSIBILITY»

To student:

Group	Name
8ВМ5И	Markus Müller

Institute	**Cybernetics**	**Department**	**Software Engineering**
Educational level	**Master**	**Educational programme**	**Computer Science and Engineering**

Initial data to «Social responsibility» chapter:	
1. *Description of work place:* – *Harmful factors in the industrial environment (meteorological conditions, harmful substances lighting, noise, vibrations, electromagnetic fields, ionizing radiation)* – *dangerous industrial factors (mechanical, thermal, electrical, etc.)* – *negative impact on the environment (atmosphere, hydrosphere, lithosphere)* *emergency situation(industrial, natural, ecological types)*	*Work place located in the Institute of Cybernetics Building*
2. *Legislative and normative documents on the topic*	*State standards, GOST, SNiP, NPB, SanPiN, federal laws*
List of tasks:	
1. *Analysis of the identified harmful factors of the industrial environment in the following sequence:* *– the physical and chemical nature of harmfulness, its relation to the topic being developed;* *– the effect of the factor on the human body;* *– reduction of permissible norms with the required dimensionality (with reference to the relevant normative and technical document);* *– proposed remedies*	Identification of all the harmful factors when researching, including physical, chemical and biological
2.Analysis of identified hazards of the industrial environment in the following sequence *– mechanical hazards (sources, means of protection;* *– thermal hazards (sources, means of protection);* *– electrical safety (including static electricity, lightning protection - sources, protective equipment);* *– fire and explosion safety (causes, preventive measures, primary means of fire extinguishing)*	Identification of the all possible hazards when researching
3.Protection of the environment: *– protection of the residential area* *– analysis of the impact of the facility on the atmosphere (emissions);* *– analysis of the impact of the object on the hydrosphere (discharges);* *– analysis of the impact of the object on the lithosphere (waste);* *– develop solutions to ensure environmental safety with references to environmental standards.*	Identification of the all possible kinds of waste when researching

4.Protection in emergency situations: *List of possible emergencies on the site;* *Choice of the most typical emergency situation;* *Development of preventive measures to prevent emergencies;* *Development of measures to improve the stability of the facility to this emergency situation;* *The development of actions as a result of the emergencies and measures to eliminate its consequences*	Identification of the all possible emergencies when researching
List of graphical data:	
Graphical plans	

Date of task obtaining	

The task was given by the consultants:

Position	Name	Academic degree	Signature	Date
Associate professor	Y.V. Anischenko	PhD		

The task was accepted by the student:

Group	Name	Signature	Date
8ВМ5И	Markus Müller		

INSTITUTE OF CYBERNETICS

TOMSK POLYTECHNIC UNIVERSITY

Master's Thesis in Informatics

On Security of Mobile Messengers

Markus Müller

Abstract

This thesis defines a list of requirements for secure mobile messengers in the form of an extended package to an existing protection profile for application software conforming to ISO/IEC 15408 (Common Criteria).

To show the applicability of this package to real world products, this work documents the security functions of the popular mobile messenger Telegram in a Common Criteria conforming security target.

For validating the assurance requirements from the extended package, the security functions from the security target are then compared with the Android version of the Telegram messenger.

The extended package, the security target for Telegram as well as the evaluation of Telegram according to the assurance activities from the extended package illustrate that the extended package for secure mobile messengers can be successfully used to evaluate the security of existing products.

Contents

1. Introduction

Every day more people use mobile messengers for communicating and exchanging sensitive information. This includes lawyers talking with their clients, doctors talking about patients and other people who are mandated by law to keep their communications confidential. At the same time, different actors such as government agencies, criminals or just upset ex-wives try to manipulate or spy on exchanged messages for different reasons.

1.1. Problem statement

As a user of mobile messengers, it is rather difficult to verify the security claims the messenger's creators made. Depending on the messenger, so far there are no or only limited independent analyses on the state of security.

Thus, the aim of this thesis is to look at the security of messengers from a holistic perspective and define universal requirements for secure messengers.

1.2. Approach Outline

Helping customers select products based on security criteria is what the Common Criteria (standardized as ISO 15408) were created for. Since the CC are broadly accepted as a framework for defining and comparing security products, this thesis will be based on the methodology and artifacts from the CC.

1.3. Sub-problems and results

For comparing and selecting the most secure messengers, I first created an implementation independent specification of security requirements (protection profile, PP) for mobile messengers. This PP defines what exactly the target of evaluation (TOE) looks like, what threats and assumptions affect the TOE and what the TOE's security objectives are.

In the next step I then defined the the security needs in an artifact called security target (ST). A ST shows how the creator of a product aims to protect the TOE (i.e. how Telegram is protected).

Similar to how products are evaluated according to the Common Methodology for Information Technology Security Evaluation (CEM), I then compared the claims from the security target and additional available guidance documents with the security requirements from the PP. In a last step, I compared the implementation of the Telegram messenger with the ST and described where Telegram does not fulfill the security requirements.

1.4. Structure of this work

To provide the reader with some basic understanding about the framework used and the importance of mobile messaging in general and especially Telegram, chapter 2 provides background information. As this work is not the first about secure mobile messaging, chapter 3 briefly describes other works in this area and highlights where this work differs. Requirements for secure mobile messengers in the form of a protection profile are defined in chapter 4. Chapter 5 describes how the security target for Telegram was created based on these requirements. An evaluation of the Android Telegram client in chapter 6 compares it with the definitions from the security target and describes where both differ. A summary in chapter 7 with the results of this work and open research questions finishes this work.

2. Background

This work is based on the framework laid out by the Common Criteria for Information Technology Security Evaluation (Common Criteria or CC) and the Common Methodology for Information Technology Security Evaluation (CEM). A brief overview over both in section 2.1.1 helps the reader understand the methodology and artifacts used in this work.

Since this work describes the security of the Telegram messenger, section 2.2 describes the concept of mobile messaging and the importance of the Telegram messenger for the Russian market.

2.1. Common Criteria and Common Evaluation Methodology

2.1.1. Common Criteria

Common Criteria is a framework in which computer system users can specify their security functional and assurance requirements (SFRs and SARs respectively) through the use of Protection Profiles (PPs), vendors can then implement and/or make claims about the security attributes of their products, and testing laboratories can evaluate the products to determine if they actually meet the claims.

Common Criteria evaluations are performed on computer security products and systems. The evaluation serves to validate claims made about the target. To be of practical use, the evaluation must verify the target's security features. The evaluation process also tries to establish the level of confidence that may be placed in the product's security features through quality assurance processes:

Target Of Evaluation (TOE) The product or system that is the subject of the evaluation.

Protection Profile (PP) a document, typically created by a user or user community, which identifies security requirements for a class of security devices relevant to that user for a particular purpose.

Product vendors can choose to implement products that comply with one or more PPs, and have their products evaluated against those PPs. In such a case, a PP may serve as a template for the product's ST (Security Target, as defined below), or the authors of the ST will at least ensure that all requirements in relevant PPs also appear in the target's ST document.

Security Target (ST) The document that identifies the security properties of the target of evaluation. The ST may claim conformance with one or more PPs. The TOE is evaluated against the Security Functional Requirements (SFRs) established in its ST. This allows vendors to tailor the evaluation to accurately match the intended capabilities of their product.

Security Functional Requirements (SFRs) Individual security functions which may be provided by a product. The Common Criteria presents a standard catalogue of such functions. The list of SFRs can vary from one evaluation to the next, even if two targets are the same type of product.

Although Common Criteria does not prescribe any SFRs to be included in an ST, it identifies dependencies where the correct operation of one function is dependent on another.

Security Assurance Requirements (SARs) Descriptions of the measures taken during development and evaluation of the product to assure compliance with the claimed security functionality. The Common Criteria provides a catalogue of these, and the requirements may vary from one evaluation to the next. The requirements for particular targets or types of products are documented in the ST and PP, respectively.

2.1.2. Common Evaluation Methodology

The Common Methodology for Information Technology Security Evaluation (CEM) is a companion document to the Common Criteria for Information Technology Security Evaluation (CC). The CEM defines the minimum actions to be performed by an evaluator in order to conduct a CC evaluation, using the criteria and evaluation evidence defined in the CC.

Each evaluation, whether of a PP or TOE (including ST), follows the same process, and has four evaluator tasks in common: the input task, the output task, the evaluation sub-activities, and the demonstration of the technical competence to the evaluation authority task.

2.2. Mobile Messaging and Telegram

Mobile messaging is a type of online chat that offers real-time text transmission over the Internet. Messages can consist of text, images or other content, depending on the messenger used.

Telegram is a free cloud-based mobile messaging service with clients for mobile and desktop systems. Users can send messages and exchange photos, videos, stickers, audio, and files of any type. Telegram also provides optional end-to-end-encrypted messaging.

Telegram is supported by Russian entrepreneur Pavel Durov. Its client-side code is open-source software but contains binary blobs, and the source code for recent versions is not always immediately published, whereas its server-side code is closed-source and proprietary.

The security of Telegram has faced notable scrutiny; critics have claimed that Telegram's security model is undermined by its use of a custom-designed encryption protocol that has not been proven reliable and secure, and by not enabling secure conversations by default. Telegram has also faced criticism for its wide-scale use by the terrorist organization Islamic State.

In February 2016, Telegram stated that it had 100 million monthly active users, sending 15 billion messages per day. According to CEO, as of April 2017, Telegram has 40 million monthly active users in Iran, surpassing other messaging apps in country.

3. Related Work

Similar to the approach of this work, the EFF created the Secure Messaging Scorecard ([EFF]), which is an overview over available messengers and a comparison of their security features. The scorecard only compares basic attributes and gives no further details how the criteria were tested.

The authors of [EMH16] provide a general overview over the market of end-to-end encrypted messaging protocols, but gives no further information about them.

In [Ung+15] the authors evaluate and systematize current secure messaging solutions and propose an evaluation framework for their security, usability, and ease-of-adoption properties. While this work provides criteria for selecting messengers, it is not suitable for verifying the security claims of security products.

Privacy and data protection mechanisms of mobile messengers are the object of [Rot+15]. Again this work is not suitable for making claims abou the security of messengers as a whole.

Other works focus on the technical side of messengers: [Ahr14] describes the Threema protocol, [JO15] analyses weaknesses in the MTProto protocol Telegram uses and [KBB17] shows a mechanism for verifying messaging protocols.

4. Requirements for Secure Mobile Messengers

For comparing mobile messengers we need a set of requirements on which we can base the comparison.

As defined by the Common Criteria, these requirements are laid out in a protection profile. Because there is already a protection profile for application software from NIAP ([Par16]), the protection profile for mobile messengers will be an extended package for NIAP's protection profile.

This chapter defines this extended package for secure mobile messengers. The method used for creation this extended package is based on the process for writing protection profiles described in [Bun10]. It differs in the way that instead of newly defining all criteria we take the existing protection profile from NIAP and refine or extend it where necessary.

First, the conformance claims and security problem are defined in sections 4.1 and 4.2. Based on this the security objectives are derived in section 4.3, which in turn are used for defining the functional and assurance requirements in sections 4.4 and 4.5. This chapter ends with an description of how the introduction for the package was created in section 4.6.

The complete extended package is provided in appendix A and will be quoted where relevant in this chapter.

4.1. Conformance Claims

The conformance claims section of a Protection Profile describes how it conforms to the Common Criteria, other protection profiles and packages. It also describes how other PPs and STs shall conform to the PP.

The PP for application software defines the baseline for security functions and assurance requirements. This EP extends the PP with additional SFRs and associated assurance activities specific to secure mobile messengers.

This EP is the first PP for mobile messengers, so we require strict conformance for any PP/ST as suggested in [Bun10].

This EP conforms to Common Criteria [CC] for Information Technology Security Evaluation, Version 3.1, Revision 5. It is CC Part 2 extended and CC Part 3 conformant. In order to be conformant to this EP, the ST must include all components in this EP and the associated App PP that are: unconditional (which are always required), selection based (which are required when certain selections are chosen in the unconditional requirements) and may include optional and/or objective components that are desirable but not required for conformance.

In accordance with CC Part 1, dependencies are not included when they are addressed by other SFRs. The assurance activities provide adequate proof that any dependencies are also satisfied.

4.2. Security Problem Definition

The security problem definition defines the security problem that is to be addressed. Since the Common Criteria describe the process of deriving the SPD as outside of scope, we follow the explanation method as defined in [Bun10].

For this we first define relevant OSPs, describe threats to the TOE and conclude with assumptions made.

4.2.1. Organizational Security Policies

The common criteria define organizational security policies as security rules, procedures, or guidelines imposed now or in the future by an actual or hypothetical organisation in the operational environment. For mobile messengers we have no such rules and neither does the base PP define any OSPs, thus the EP defines no OSPs.

4.2.2. Threats

To define threats we need to define what happens if we do not have a TOE or the TOE provides no security at all. This task can be subdivided into three parts: first, we need to describe what the assets are. Base on this we can define what the adverse actions for this assets might be and who the threat agents are.

The assets a secure mobile messenger needs to protect are the content of the conversations, both during transmission and while stored on the device, as well as the identities and authenticity of the users.

External entities that potentially may attack the TOE are called threat agents. They satisfy one or more of the following criteria:

- External entities not authorized to access assets may attempt to access them either by masquerading as an authorized entity or by attempting to use TSF services without proper authorization.
- External entities authorized to access certain assets may attempt to access other assets they are not authorized to either by misusing services they are allowed to use or by masquerading as a different external entity.
- Untrusted subjects may attempt to access assets they are not authorized to either by misusing services they are allowed to use or by masquerading as a different subject.

From this we can define the following groups of threats to mobile messengers:

- Active and passive attacks on the network which threatens the transmission of messages.
- Other software on the same same device might try to access the assets.
- Attackers who get physical access to the device may try to access the assets.
- Attackers try to impersonate other users or try to steal their identities.
- Insecure design or implementation weaknesses may enable an attacker to access the assets.

The NIAP PP defines the following threats for application software, which already address the part of the problems outlined before.

- **T.NETWORK_ATTACK** An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may engage in communications with the application software or alter communications between the application software and other endpoints in order to compromise it.
- **T.NETWORK_EAVESDROP** An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may monitor and gain access to data exchanged between the application and other endpoints.
- **T.LOCAL_ATTACK** An attacker can act through unprivileged software on the same computing platform on which the application executes. Attackers may provide maliciously formatted input to the application in the form of files or other local communications.
- **T.PHYSICAL_ACCESS** An attacker may try to access sensitive data at rest.

These threats already address the first three groups of threats described above. To address the other two groups, we defined the following additional threats:

- **T.UNSAFE_AUTHFACTOR_VERIFICATION** An attacker can take advantage of an unsafe method for performing verification of an authorization factor (e.g. SMS), resulting in exposure of the cryptographic material or user data.
- **T.SECURITY_FUNCTIONALITY_FAILURE** A component of the mobile messenger (e.g. random number generator) may fail during start-up or during operations causing a compromise or failure in the security functionality of the messenger, leaving it vulnerable to attackers.

4.2.3. Assumptions

The base PP defines the following assumptions:

- **A.PLATFORM** The TOE relies upon a trustworthy computing platform for its execution. This includes the underlying platform and whatever runtime environment it provides to the TOE.
- **A.PROPER_USER** The user of the application software is not willfully negligent or hostile, and uses the software in compliance with the applied enterprise security policy.
- **A.PROPER_ADMIN** The administrator of the application software is not careless, willfully negligent or hostile, and administers the software within compliance of the applied enterprise security policy.

These assumptions already cover what we assume about our environment, so the EP adds no further assumptions.

4.3. Security Objectives

4.3.1. Deriving the Security Objectives

Our security problem definition for mobile messengers consists of the threats described in the preceding section. To define a secure TOE and an operational environment that will counter these threats, we need to answer three questions: First, we define where the TOE will be placed and if it can be physically attacked there. Then we define the purpose of the TOE and finally we say how the TOE will be managed.

Location

A mobile messenger is usually installed on a portable device such as a phone or tablet. A loss of physical control over the device may result in the loss of confidentiality.

Sensitive data on the device should therefore be encrypted and the application should be able to restrict the access to it. The first requirements is already part of the [App PP] (O.PROTECTED_STORAGE), so this extended package need to define a security objective for the second one (O.ACCESS).

O.PROTECTED_STORAGE To address the issue of loss of confidentiality of user data in the event of loss of physical control of the storage medium, conformant TOEs will use data-at-rest protection. This involves encrypting data and keys stored by the TOE in order to prevent unauthorized access to this data. This also includes unnecessary network communications whose consequence may be the loss of data.

O.ACCESS To address issues associated with the loss of confidentiality of user data in the event of loss of physical control of the device, conformant messengers implement mechanisms to restrict access to them. This includes automatic and manual access restrictions.

Purpose

A mobile messenger should securely send messages sent between communicating parties and it should prevent confidential data to be leaked or messages to be corrupted. At the same time it should guarantee the authenticity of the users as well as the authenticity of the exchanged messages.

The first requirement again is already covered in the [App PP] (*O.PROTECTED_-COMMS*) and for the second requirement this extended package defines an additional objective (*O.AUTHENTICATION*).

O.PROTECTED_COMMS To address both passive (eavesdropping) and active (packet modification) network attack threats, conformant TOEs will use a trusted channel for sensitive data. Sensitive data includes cryptographic keys, passwords, and any other data specific to the application that should not be exposed outside of the application.

O.AUTHENTICATION To address issues associated with weak authentication mechanisms (e.g. identity theft, impersonation of other users), the messenger must ensure that it protects the authentication process by using strong authentication mechanisms (e.g. two-factor authentication).

As secure mobile messengers are application software in the sense of the [App PP], the objectives regarding integrity (*O.INTEGRITY*) and quality (*O.QUALITY*) defined in it also apply.

O.INTEGRITY Conformant TOEs ensure the integrity of their installation and update packages, and also leverage execution environment-based mitigations. Software is seldom if ever shipped without errors, and the ability to deploy patches and updates to fielded software with integrity is critical to enterprise network security. Processor manufacturers, compiler developers, execution environment vendors, and operating system vendors have developed execution environment-based mitigations that increase the cost to attackers by adding complexity to the task of compromising systems. Application software can often take advantage of these mechanisms by using APIs provided by the runtime environment or by enabling the mechanism through compiler or linker options.

O.QUALITY To ensure quality of implementation, conformant TOEs leverage services and APIs provided by the runtime environment rather than implementing their own versions of these services and APIs. This is especially important for cryptographic services and other complex operations such as file and media parsing. Leveraging this platform behavior relies upon using only documented and supported APIs.

Management

Secure mobile messengers are managed by users and the enterprise. As described in the [App PP], applications should provide consistent and supported interfaces for this, so this extended package does not define any further objectives.

O.MANAGEMENT To facilitate management by users and the enterprise, conformant TOEs provide consistent and supported interfaces for their security-relevant configuration and maintenance. This includes the deployment of applications and application updates through the use of platform-supported deployment mechanisms and formats, as well as providing mechanisms for configuration. This also includes providing control to the user regarding disclosure of any PII.

4.4. Security Functional Requirements

This section describes how the security functional requirements are derived from the security objectives for the TOE.

4.4.1. TSF- and User-initiated locked state (FTA_SSL_EXT.1)

To prevent attackers to access sensitive data, the messenger needs to be able to lock itself. This should be either when the user requests it or after a defined timeout. When going into the locked mode, the messenger should be able to do a defined operation (e.g. close connections).

FTA_SSL_EXT.1.1 The TSF shall transition to a locked state after a time interval of inactivity and a user initiated lock, and upon transitioning to the locked state, the TSF shall perform the following operations: *[assignment: actions performed upon transitioning to the locked state]*.
Assurance Activity: The evaluator shall perform the following tests:

- Test 1: The evaluator unlocks the application and waits for the defined interval of inactivity. Then the evaluator shall verify that the application transitioned into the locked state and the defined actions were executed.
- Test 2: The evaluator should initiate a lock according to the procedures described in the documentation and shall verify that the defined actions were executed.

To detect and complicate malicious unlocking attempts, the messenger should detect them and execute some defined action.

FTA_SSL_EXT.1.2 The TSF shall detect when *[assignment: range of acceptable values]* of unsuccessful unlocking attempts occur related to last successful unlocking by that user. When the defined number of unsuccessful unlocking attempts has been [selection: met, surpassed], the TSF shall perform *[assignment: action to execute]*.
Assurance Activity: The evaluator shall try to unlock the application using procedures described in the documentation for the defined number of times and verify that the application performs the defined action.

To prevent shoulder surfing while unlocking the messenger, it should only show the input in an obscured way.

FTA_SSL_EXT.1.3 The TSF shall provide only obscured feedback to the device's display to the user while the unlocking is in progress.
Assurance Activity: The evaluator shall unlock the application and verify that the output is obscured.

4.4.2. Protection of Data in Transit (FTP_DIT_EXT.1)

For this SFR we need to refine the protocols, so that special encryption modes used by different messengers are allowed.

The application note for FTP_DIT_EXT.1 from the [App PP] says that extended packages may override this requirement.

Application Note: Extended packages may override this requirement to provide for other protocols. Encryption is not required for applications transmitting data that is not sensitive.

The refinement removes the options of not sending data, which are contradicting the point of messengers. Furthermore it overwrites the use of specific protocols and looks as follows:

FTP_DIT_EXT.1.1 The application shall [selection:

- encrypt all transmitted sensitive data with **[assignment: encryption scheme]**,
- encrypt all transmitted data with **[assignment: encryption scheme]**

] between itself and another trusted IT product.
Application Note: Extended packages may override this requirement to provide for other protocols. Encryption is not required for data that is not sensitive.
Assurance Activity: The evaluator shall exercise the application (attempting to transmit data) while capturing packets from the application. The evaluator shall review the packet capture and verify that no sensitive data is transmitted in the clear.

4.4.3. Timing of Authentication (FIA_UAU_EXT.1)

The messenger should make sure that for all security relevant functions the user needs to be authenticated. Only certain exceptions (e.g. for registration with the service) should be allowed before authentication.

FIA_UAU_EXT.2.1 The TSF shall allow [selection: *[assignment: list of actions]*, no actions] on behalf of the user to be performed before the user is authenticated.
Assurance Activity: The evaluator shall use the application before authentication and verify that the list of defined actions can be used.

FIA_UAU_EXT.2.2 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.
Assurance Activity: The evaluator shall use the application before authentication and verify that no other than the defined actions can be used.

4.4.4. Multiple authentication mechanisms (FIA_UAU.5)

The messenger should provide multiple authentication mechanisms (e.g. SMS and password) to protect the user's identity.

FIA_UAU.5.1 The TSF shall provide *[assignment: list of multiple authentication mechanisms]* to support user authentication.
Assurance Activity: The evaluator shall authenticate with the application and shall verify that all defined authentication mechanisms are supported.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to *[assignment: the rules describing how the multiple authentication mechanisms provide authentication]*.
Assurance Activity: The evaluator shall authenticate with the application using procedures described in the documentation and shall verify that the defined rules conform with the documentation.

4.4.5. Re-Authentication (FIA_UAU.6)

Based on certain conditions (e.g. when a user manages and terminates an active session) the user should re-authenticate.

FIA_UAU.6.1 The TSF shall re-authenticate the user under the conditions *[assignment: list of conditions under which re-authentication is required]*.
Assurance Activity: The evaluator shall cause the defined conditions and shall verify that the user needs to be re-authenticated.

4.4.6. Protected Authentication Feedback (FIA_UAU.7)

As with the passcode unlocking, the input of authentication data should be obscured to prevent shoulder surfing.

FIA_UAU.7.1 The TSF shall provide only [obscured feedback to the device's display] to the user while the authentication is in progress.
Assurance Activity: The evaluator shall authenticate using each available authentication method and shall verify that the output is obscured.

4.5. Security Assurance Requirements

The assurance requirements from the [App PP] are already sufficient, which is why no additional requirements are added to this extended package.

4.6. Protection Profile Introduction

This section describes how the introduction to the PP was derived.

Section B.4 of the [CC] part 1 describes the introduction for a PP as follows:

> The PP introduction describes the TOE in a narrative way on two levels of abstraction:
>
> A the PP reference, which provides identification material for the PP;
> B the TOE overview, which briefly describes the TOE.

The reference is straightforward and looks as follows:

> - PP Reference: Extended Package for Secure Mobile Messengers
> - PP Version: 1.0
> - PP Date: 01-Jun-2017

The second requirement for the PP introduction, the overview over the TOE, is split into two parts, similar to how the [App PP] defines it.

The TOE overview gives a short description over what the EP is about:

> This Extended Package defines requirements for the evaluation of Secure Mobile Messengers.
>
> Such products are generally mobile software applications designed to conduct confidential and authenticated conversations over untrusted networks (i.e. the Internet).
>
> Audio and video communications are outside of the scope of the current version of this PP, but are expected to be included in the scope of the next version.

In addition to this the use cases describe how a messenger can be used:

> Secure Mobile Messengers perform tasks associated primarily with the following use case.
>
> **Sending and receiving messages** The application allows a user to communicate interactively and non-interactively with one or more other users over a secure channel. This includes exchanging text messages, media files and other types of information such as geological positions or contact information.

5. Security Target for Telegram

This chapter describes how the security target (ST) for the Telegram messenger was derived.

The process for creating a ST is similar to the process of creating a PP, so this chapter is again based on the guidelines described in [Bun10].

5.1. Writing the conformance claims

As described in [12] section A.5, the conformance claims section must include a description of how the ST conforms to the CC, to Protection Profiles and to packages.

The ST for Telegram is conformant on the application software protection profile ([Par16]) and its extended package for secure mobile messengers defined in appendix A). The ST for Telegram does not change any SFRs or SARs, so it can claim strict conformance.

> This Security Target is CC Part 2 extended and CC Part 3 conformant. It claims conformance to the following Protection Profile:
>
> - Application Software Protection Profile (App PP). Version 1.2 as of 2016; strict conformance.
> - Secure Mobile Messenger Extended Package (SMM EP). Version 1.0 as of 2017; strict conformance.
>
> Common Criteria [CC] version 3.1 revision 5 is the basis for this conformance claim.

5.2. Determining the security problem definition

If a ST is for a more specific TOE than the TOE from the PP, it can add extend the security problem definition. Since the EP for secure mobile messengers already provides all necessary extensions for messengers, Telegram does not have to extend it any further. The security problem definition for Telegram is therefore the same as in the extended package from appendix A, which is why we just quote it and refer to the source.

> **Threats** This section identifies the threats against the TOE. These threats have been taken from the [App PP] and [SMM EP].
>
> . . .
>
> **Organisational security policies (OSPs)** Neither [App PP] nor [SMM EP] define organizational security policies.
>
> **Assumptions** The following assumptions have been taken from the [App PP]. The [SMM EP] defines no further assumptions.
>
> . . .

5.3. Deriving security objectives

Deriving security objectives for the ST is the same as for the extended package. In the ST we therefore just refer to the PP and EP the ST is based on for further details.

> This section identifies the security objectives of the TOE and its supporting environment. The security objectives identify the responsibilities of the TOE and its environment in meeting the security needs.
>
> The complete security problem definition may be found in [App PP] and [SMM EP] and this section reproduces only the corresponding security objectives for for reader convenience. The [App PP] and [SMM EP] offer additional information about the identified security objectives, but that has not been reproduced here and both the [App PP] and the [SMM EP] should be consulted if there is interest in that material.

5.4. Deriving the SFRs

For deriving the SFRs for Telegram we need to complete all assignment and selection operations from the PP and EP. The following sections describe why which value was selected. SFRs without assignment or selection operations are included in the ST, but excluded from the following descriptions.

5.4.1. Cryptographic Support (FCS)

Random Bit Generation (FCS_RBG)

As Telegram does not implement any random bit generator, it relies on the platform to provide this functionality.

Random Bit Generation (FCS_RBG_EXT.1)
FCS_RBG_EXT.1.1 The application shall invoke platform-provided DRBG functionality for its cryptographic operations.

Storage of Credentials (FCS_STO)

Telegram does not store any credentials in non-volatile memory.

Storage of Credentials (FCS_STO_EXT.1)
FCS_STO_EXT.1.1 The application shall [not store any credentials] to non-volatile memory.

5.4.2. User Data Protection (FDP)

Platform Resources (FDP_DEC)

For connecting to the Telegram service, Telegram obviously needs network connectivity. Since Telegram can send voice messages, pictures and locations, it also needs acces to those services.

For finding other users, Telegram also needs access to the address book.

Access to Platform Resources (FDP_DEC_EXT.1)
FDP_DEC_EXT.1.1 The application shall restrict its access to network connectivity, camera, microphone and location services.
FDP_DEC_EXT.1.2 The application shall restrict its access to address book.

Network Communications (FDP_NET)

Telegram authenticates with the Telegram service and exchanges messages only with it.

Network Communications (FDP_NET_EXT.1)
FDP_NET_EXT.1.1 The application shall restrict network communication to *user authentication and exchanging messages*.

Application Data (FDP_DAR)

As Telegram does not encrypt any files on its own, it uses Android to protect stored data. Apart from messages and files it stores no sensitive data.

Encryption Of Sensitive Application Data (FDP_DAR_EXT.1)
FDP_DAR_EXT.1.1 The application shall not store any sensitive data in non-volatile memory.

5.4.3. Security Management (FMT)

Specification of Management Functions (FMT_SMF)

Telegram's settings menu provides functions to manage two-step authentication, sessions as well as locking and passcode options.

Specification of Management Functions (FMT_SMF.1)
FMT_SMF.1.1 The TSF shall be capable of performing the following management functions [

- *managing two-step authentication,*
- *managing active sessions,*
- *define time-out interval for automatic transition into the locked state,*
- *change passcode for unlocking the TOE*

].

5.4.4. Privacy (FPR)

Personally Identifiable Information (FPR_ANO)

Telegram requires the user to provide his first name and an optional last name for using the service.

User Consent for Transmission of PII (FPR_ANO_EXT.1)
FPR_ANO_EXT.1.1 The application shall require user approval before executing *setting or changing the user's name*.

5.4.5. Protection of the TSF (FPT)

Anti-Exploitation (FPT_AEX)

Telegram is not using any memory with read and write permissions.

Anti-Exploitation Capabilities (FPT_AEX_EXT.1)
FPT_AEX_EXT.1.2 The application shall not allocate any memory region with

both write and execute permissions.

Installation and Update (FPT_TUD)

Telegram uses the Google Play store for installing and updating and therefore does not implement its own mechanisms.

Integrity for Installation and Update (FPT_TUD_EXT.1)

FPT_TUD_EXT.1.1 The application shall leverage the platform to check for updates and patches to the application software.

FPT_TUD_EXT.1.5 The application shall leverage the platform to query the current version of the application software.

Third Party Libraries (FPT_LIB)

Telegram bundles some third party libraries with its source, but distributes the native code as one library (*libtmessages*).

Use of Third Party Libraries (FPT_LIB_EXT.1)

FPT_LIB_EXT.1.1 The application shall be packaged with only *[*

- boringssl in version 3.2.6
- breakpad in version 3.2.6
- ffmpeg in version 3.4.2
- libjpeg in version 3.2.6
- libwebp in version 3.2.6
- libyuv in version 3.10.1
- opus in version 3.2.6
- sqlite in version 3.9.0

].

5.4.6. TOE Access (FTA)

Session Locking (FTA_SSL)

Telegram has a passcode for locking the application. It does not execute any special operations when locking and it does not do anything when the passcode is entered too often.

TSF- and User-initiated locked state (FTA_SSL_EXT.1)

FTA_SSL_EXT.1.1 The TSF shall transition to a locked state after a time interval of inactivity and a user initiated lock, and upon transitioning to the locked state, the TSF shall perform the following operations: *[no action]*.

FTA_SSL_EXT.1.2 The TSF shall detect when *[10]* of unsuccessful unlocking attempts occur related to last successful unlocking by that user. When the defined number of unsuccessful unlocking attempts has been [met], the TSF shall perform *[no action]*.

5.4.7. Trusted Path/Channel (FTP)

Data in Transit (FTP_DIT)

Telegram encrypts all its communication with a custom encryption protocol (MTProto).

Protection of Data in Transit (FTP_DIT_EXT.1)

FTP_DIT_EXT.1.1 The application shall encrypt all transmitted sensitive data with *a custom encryption scheme* between itself and another trusted IT product.

5.4.8. Identification and Authentication (FIA)

User Authentication (FIA_UAU)

For registering with Telegram, the user needs to be able to do so before he authenticates.

Timing of Authentication (FIA_UAU_EXT.1)

FIA_UAU_EXT.1.1 The TSF shall allow *registration with the service* on behalf of the user to be performed before the user is authenticated.

Authenticating with the Telegram service works via SMS. The user receives a random number and provides this number to Telegram. In addition to this the user can define a custom password as a second authentication factor.

Multiple authentication mechanisms (FIA_UAU.5)

FIA_UAU.5.1 The TSF shall provide *SMS token with additional user-defined password* to support user authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to *the verification of the number received via SMS and a user-defined password*.

Telegram users need to be re-authenticated when the user's session is terminated (e.g. by using the "Active Sessions" option) or expired (e.g. when the account is deleted).

Re-Authentication (FIA_UAU.6)

FIA_UAU.6.1 The TSF shall re-authenticate the user under the conditions *session termination, session expiration*.

5.5. Defining the SARs

The SARs for Telegram are the same as in the application software protection profile and the extended package, to which the ST is conforming. Therefore we just refer to those documents and do not copy them into the ST.

5.6. Defining the TOE summary specification

The TOE summary specification (TSS) provides a description of how the TOE satisfies all the SFRs. Based on the SFRs from preceding chapter, the TSS was derived by analyzing the Telegram application and parts of its source code.

5.7. Writing the ST introduction

The ST introduction is similar to the introduction of an PP. It only adds the TOE description, an additional part about the physical and logical scope of the TOE.

The physical boundary for the TOE is the Telegram client application with the required configuration from chapter C. The Telegram service itself is not part of the TOE.

The logical boundary in section B.1.3 is drawn around the categories used for the SFRs and TSS. Each section provides a high-level overview over Telegrams security features.

6. Technical Evaluation

To validate that the assurance requirements from chapter A are realistic and work with a real product, we evaluate them for the Telegram messenger.

Whenever the assurance activity refers to documentation, appendix C provides the necessary parts.

The evaluated configuration consists of two instances of two instances of Telegram installed on two different devices. Both instances are configured conforming to appendix C.

6.1. Assurance Activities Report

6.1.1. FCS_RBG_EXT.1 Random Bit Generation Services

FCS_RBG_EXT.1.1

SFR The application shall invoke platform-provided DRBG functionality for its cryptographic operations.

Assurance Activity The evaluator shall verify that the application uses at least one of *javax.crypto.KeyGenerator* class or the *java.security.SecureRandom* class or */dev/random* or */dev/urandom*.

Results As described in the assurance activity of the [App PP], I decompiled the application binary and looked for the required APIs in the output.

```
org/telegram/messenger/Utilities.smali:
const-string/jumbo v4, "/dev/urandom"
```

6.1.2. FCS_STO_EXT.1 Storage of Credentials

FCS_STO_EXT.1.1

SFR The application shall [not store any credentials] to non-volatile memory.

Assurance Activity None.

Results The assurance activity of the [App PP] requires only actions when the application implement its own functionality or invokes platform-provided functionality to securely store credentials. Telegram does not store credentials in non-volatile memory.

6.1.3. FDP_DEC_EXT.1 Access to Platform Resources

FDP_DEC_EXT.1.1

SFR The application shall restrict its access to network connectivity, camera, microphone and location services.

Assurance Activity The evaluator shall inspect permissions presented at installation time (Android 5.1 and below) or on-access (Android 6.0 and above) for each hardware resource an app intends to access.

Results Telegram asks for the following permissions at installation time: Location, SMS, phone, camera, microphone, Wi-Fi connection information, Device ID & call information

FDP_DEC_EXT.1.2

SFR The application shall restrict its access to address book.

Assurance Activity The evaluator shall inspect permissions presented at installation time (Android 5.1 and below) or on-access (Android 6.0 and above) for each sensitive information repository an app intends to access.

Results Telegram asks for the following permissions installation time: Identity, contacts, photos/media/files, Device ID & call information

6.1.4. FDP_NET_EXT.1 Network Communications

FDP_NET_EXT.1.1

SFR The application shall restrict network communication to *user authentication and exchanging messages*.

Assurance Activity The evaluator shall run the application. While the application is running, the evaluator shall sniff network traffic ignoring all non-application associated traffic and verify that any network communications witnessed are documented in the TSS or are user-initiated.

Results All connections are documented or user-initiated.

6.1.5. FDP_DAR_EXT.1 Encryption Of Sensitive Application Data

FDP_DAR_EXT.1.1

SFR The application shall not store any sensitive data.

Assurance Activity None.

Results Not applicable.

6.1.6. FMT_MEC_EXT.1 Supported Configuration Mechanism

FMT_MEC_EXT.1.1

SFR The application shall invoke the mechanisms recommended by the platform vendor for storing and setting configuration options.

Assurance Activity The evaluator shall run the application and make security-related changes to its configuration. The evaluator shall check that at least one XML file at location */data/data/package/shared_prefs/* reflects the changes made to the configuration to verify that the application used *SharedPreferences* and/or *PreferenceActivity* classes for storing configuration data, where package is the Java package of the application.

Results Telegram is using *SharedPreferences* and stores the configuration in file $/data/data/org.telegram.messenger/shared_prefs/userconfing.xml$.

6.1.7. FMT_CFG_EXT.1 Secure by Default Configuration

FMT_CFG_EXT.1.1

SFR The application shall provide only enough functionality to set new credentials when configured with default credentials or no credentials.

Assurance Activity The evaluator shall check the TSS to determine if the application requires any type of credentials and if the application installs with default credentials. If the application uses any default credentials the evaluator shall run the following tests.

Test 1 The evaluator shall install and run the application without generating or loading new credentials and verify that only the minimal application functionality required to set new credentials is available.

Test 2 The evaluator shall attempt to clear all credentials and verify that only the minimal application functionality required to set new credentials is available.

Test 3 The evaluator shall run the application, establish new credentials and verify that the original default credentials no longer provide access to the application.

Results

Test 1 After an informational screen about the functionality of Telegram, the application presents only a authentication screen.

Test 2 After a log out, the same informational screen about the functionality of Telegram and a authentication screen are shown.

Test 3 After a log out, the old identification is not available on the device anymore, until the user authenticates again.

FMT_CFG_EXT.1.2

SFR The application shall be configured by default with file permissions which protect it and its data from unauthorized access.

Assurance Activity The evaluator shall run $ls - alR|grep - E'\cdot......(r| - w| - -x)'$ inside the application's data directories to ensure that all files are not world-accessible (either read, write, or execute). The command should not print any files. The evaluator shall also verify that no sensitive data is written to external storage as this data can be read/modified by any application containing the $READ\_EXTERNAL\_STORAGE$ and/or $WRITE\_EXTERNAL\_STORAGE$ permissions.

Results

6.1.8. FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1

SFR The TSF shall be capable of performing the following management functions enable/disable the transmission of any PII, *managing two-step authentication, managing active sessions, define time-out interval for automatic transition into the locked state, change passcode for unlocking the TOE*.

Assurance Activity The evaluator shall verify that every management function mandated by the PP is described in the operational guidance and that the description contains the information required to perform the management duties associated with the management function. The evaluator shall test the application's ability to provide the management functions by configuring the application and testing each option selected from above. The evaluator is expected to test these functions in all the ways in which the ST and guidance documentation state the configuration can be managed.

Results The management functions are documented and work as described.

6.1.9. FPR_ANO_EXT.1 User Consent for Transmission of Personally Identifiable Information

FPR_ANO_EXT.1.1

SFR The application shall require user approval before executing *setting or changing the user's name*.

Assurance Activity The evaluator shall inspect the TSS documentation to identify functionality in the application where PII can be transmitted, and perform the following tests.

Test 1 The evaluator shall run the application and exercise the functionality responsibly for transmitting PII and verify that user approval is required before transmission of the PII.

Results When registering, Telegram asks the user for his first name and an optional last name.

6.1.10. FPT_API_EXT.1 Use of Supported Services and APIs

FPT_API_EXT.1.1

SFR The application shall use only documented platform APIs.

Assurance Activity The evaluator shall verify that the TSS lists the platform APIs used in the application. The evaluator shall then compare the list with the supported APIs (available through e.g. developer accounts, platform developer groups) and ensure that all APIs listed in the TSS are supported.

Results All listed APIs are supported.

6.1.11. FPT_AEX_EXT.1 Anti-Exploitation Capabilities

FPT_AEX_EXT.1.1

SFR The application shall not request to map memory at an explicit address except for *no exception*.

Assurance Activity The evaluator shall run the same application on two different Android systems. Connect via ADB and inspect */proc/PID/maps*. Ensure the two different instances share no mapping locations.

Results The two devices share no mapping locations.

```
$ adb -s emulator-5556 shell
generic_x86:/ # ps | grep telegram | cut -d\ -f5
3925
generic_x86:/ # cat /proc/3925/maps
...

$ comm -1 -2 maps_device1 maps_device2
b6cfc000-b6cfd000 rw-p 00000000 00:00 0 [anon:linker_alloc_small_objects]
```

FPT_AEX_EXT.1.2

SFR The application shall [selection: not allocate any memory region with both write and execute permissions, allocate memory regions with write and execute permissions for only *[assignment: list of functions performing just-in-time compilation]*].

Assurance Activity The evaluator shall perform static analysis on the application to verify that *mmap* is never invoked with both the *PROT_WRITE* and *PROT_EXEC* permissions, and *mprotect* is never invoked.

Results *mprotect* is never invoked and *mmap* is never invoked with both write and exec permissions.

```
$ ack "(mprotect|mmap) ?\(" *
TMessagesProj/jni/breakpad/common/memory.h
119: void *a = sys_mmap(NULL, page_size_ * num_pages, PROT_READ | \
       PROT_WRITE,

TMessagesProj/jni/breakpad/common/linux/memory_mapped_file.cc
92: void* data = sys_mmap(NULL, file_len, PROT_READ, MAP_PRIVATE, fd, \
     offset);
```

FPT_AEX_EXT.1.3

SFR The application shall be compatible with security features provided by the platform vendor.

Assurance Activity The evaluator shall ensure that the application can successfully run on the latest version of Android.

Results Telegram was tested on Android 7.1, the newest version of Android available at the time of writing.

FPT_AEX_EXT.1.4

SFR The application shall not write user-modifiable files to directories that contain executable files unless explicitly directed by the user to do so.

Assurance Activity The evaluator shall run the program, mimicking normal usage, and note where all files are written. The evaluator shall ensure that there are no executable files stored under */data/data/package/* where package is the Java package of the application.

Results Telegram stores no executable files under $/data/data/org.telegram.messenger$.

```
generic_x86:/ # find /data/data/org.telegram.messenger -type f -perm /111
generic_x86:/ #
```

FPT_AEX_EXT.1.5

SFR The application shall be compiled with stack-based buffer overflow protection enabled.

Assurance Activity Applications that are entirely Java run in the Java machine and do not need traditional stack protection. For applications using Java Native Interface (JNI), the evaluator shall ensure that the $-fstack-protector-strong$ or $-fstack-protector-all$ flags are used. The $-fstack-protector-all$ flag is preferred but $-fstack-protector-strong$ is acceptable.

Results Telegram uses native code, but it compiles it without any Stack protection activated.

```
$ grep fstack-protector Telegram/TMessagesProj/jni/Android.mk
$
```

6.1.12. FPT_TUD_EXT.1 Integrity for Installation and Update

FPT_TUD_EXT.1.1

SFR The application shall leverage the platform to check for updates and patches to the application software.

Assurance Activity The evaluator shall check for an update using procedures described in the documentation and verify that the application does not issue an error. If it is updated or if it reports that no update is available this requirement is considered to be met.

Results Telegram is distributed using the Google Play store. During the test no update was available.

FPT_TUD_EXT.1.2

SFR The application shall be distributed using the format of the platform-supported package manager.

Assurance Activity The evaluator shall ensure that the application is packaged in the Android application package (APK) format.

Results Telegram is packaged in the APK format.

```
generic_x86:/ # find / -name "*apk" | grep telegram
/data/app/org.telegram.messenger-1/base.apk
```

FPT_TUD_EXT.1.3

SFR The application shall be packaged such that its removal results in the deletion of all traces of the application, with the exception of configuration settings, output files, and audit/log events.

Assurance Activity The evaluator shall record the path of every file on the entire filesystem prior to installation of the application, and then install and run the application. Afterwards, the evaluator shall then uninstall the application, and compare the resulting filesystem to the initial record to verify that no files, other than configuration, output, and audit/log files, have been added to the filesystem.

Results No additional files have been added to the filesystem.

FPT_TUD_EXT.1.4

SFR The application shall not download, modify, replace or update its own binary code.

Assurance Activity The evaluator shall install the application and then locate all of its executable files. The evaluator shall then, for each file, save off either a hash of the file or a copy of the file itself. The evaluator shall then run the application and exercise all features of the application as described in the TSS. The evaluator shall then compare each executable file with the either the saved hash or the saved copy of the files. The evaluator shall verify that these are identical.

Results All files are identical and were not changed.

FPT_TUD_EXT.1.5

SFR The application shall leverage the platform to query the current version of the application software.

Assurance Activity The evaluator shall query the application for the current version of the software according to the operational user guidance (AGD_OPE.1) and shall verify that the current version matches that of the documented and installed version.

Results The versions match.

FPT_TUD_EXT.1.6

SFR The application installation package and its updates shall be digitally signed such that its platform can cryptographically verify them prior to installation.

Assurance Activity The evaluator shall verify that the TSS identifies how the application installation package and updates to it are signed by an authorized source. The definition of an authorized source must be contained in the TSS. The evaluator shall also ensure that the TSS (or the operational guidance) describes how candidate updates are obtained.

Results The description is complete.

6.1.13. FPT_LIB_EXT.1 Use of Third Party Libraries

FPT_LIB_EXT.1.1

SFR The application shall be packaged with only *[*

- boringssl in version 3.2.6
- breakpad in version 3.2.6
- ffmpeg in version 3.4.2
- libjpeg in version 3.2.6
- libwebp in version 3.2.6
- libyuv in version 3.10.1
- opus in version 3.2.6
- sqlite in version 3.9.0

].

Assurance Activity The evaluator shall install the application and survey its installation directory for dynamic libraries. The evaluator shall verify that libraries found to be packaged with or employed by the application are limited to those in the assignment.

Results The installed version of Telegram provides only one shared library named *libtmessages*.

6.1.14. FTP_DIT_EXT.1 Protection of Data in Transit

SFR The application shall encrypt all transmitted sensitive data with *a custom encryption scheme* between itself and another trusted IT product.

Assurance Activity The evaluator shall exercise the application (attempting to transmit data) while capturing packets from the application. The evaluator shall review the packet capture and verify that no sensitive data is transmitted in the clear.

Results No sensitive data was transmitted in clear.

6.1.15. FTA_SSL_EXT.1 TSF- and User-initiated locked state

FTA_SSL_EXT.1.1

SFR The TSF shall transition to a locked state after a time interval of inactivity and a user initiated lock, and upon transitioning to the locked state, the TSF shall perform the following operations: *[no action]*.

Assurance Activity The evaluator shall perform the following tests:

Test 1 The evaluator unlocks the application and waits for the defined interval of inactivity. Then the evaluator shall verify that the application transitioned into the locked state and the defined actions were executed.

Test 2 The evaluator should initiate a lock according to the procedures described in the documentation and shall verify that the defined actions were executed.

Results The application transitions into the locked state as defined. As there are no actions to be executed, no further execution was noticed.

FTA_SSL_EXT.1.2

SFR The TSF shall detect when *[10]* of unsuccessful unlocking attempts occur related to last successful unlocking by that user. When the defined number of unsuccessful unlocking attempts has been [met], the TSF shall perform *[no action]*.

Assurance Activity The evaluator shall try to unlock the application using procedures described in the documentation for the defined number of times and verify that the application performs the defined action.

Results As there is no defined action that should happen, the passcode can be tried unlimited times.

FTA_SSL_EXT.1.3

SFR The TSF shall provide only obscured feedback to the device's display to the user while the unlocking is in progress.

Assurance Activity The evaluator shall unlock the application and verify that the output is obscured.

Results When typing in the passcode, the letters are obfuscated after a short period of visibility (about one second).

6.1.16. FIA_UAU_EXT.1 Timing of Authentication

FIA_UAU_EXT.1.1

SFR The TSF shall allow *registration with the service* on behalf of the user to be performed before the user is authenticated.

Assurance Activity The evaluator shall use the application before authentication and verify that the list of defined actions can be used.

Results The user can register with Telegram before authenticating.

FIA_UAU_EXT.1.2

SFR The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Assurance Activity The evaluator shall use the application before authentication and verify that no other than the defined actions can be used.

Results Before authenticating no other options than registering with the Telegram service can be used.

6.1.17. FIA_UAU.4 Multiple authentication mechanisms

FIA_UAU.5.1

SFR The TSF shall provide *SMS token with additional user-defined password* to support user authentication.

Assurance Activity The evaluator shall authenticate with the application and shall verify that all defined authentication mechanisms are supported.

Results Both authentication mechanisms are available.

FIA_UAU.5.2

SFR The TSF shall authenticate any user's claimed identity according to *the verification of the number received via SMS and a user-defined password*.

Assurance Activity The evaluator shall authenticate with the application using procedures described in the documentation and shall verify that the defined rules conform with the documentation.

Results Both authentication mechanisms work as described.

6.1.18. FIA_UAU.6 Re-Authentication

FIA_UAU.6.1

SFR The TSF shall re-authenticate the user under the conditions *session termination, session expiration*.

Assurance Activity The evaluator shall cause the defined conditions and shall verify that the user needs to be re-authenticated.

Results The re-authentication works as described.

6.1.19. FIA_UAU.7 Protected Authentication Feedback

FIA_UAU.7.1

SFR The TSF shall provide only *obscured feedback to the device's display* to the user while the authentication is in progress.

Assurance Activity The evaluator shall authenticate using each available authentication method and shall verify that the output is obscured.

Results Typed in characters from the SMS are not obfuscated. The additional password is obfuscated.

7. Summary and conclusions

In this work I created list of requirements for secure mobile messengers in the form of an extended package to the application software protection profile according to the Common Criteria framework. To show the applicability of this package to real world products, I then described the security functions of a popular mobile messenger, Telegram, and documented it in a Common Criteria conforming security target. For validating the assurance requirements from the extended package, I then compared the stated security functions with the implemented versions of the Telegram messenger app.

The extended package, the security target for Telegram as well as the evaluation of Telegram according to the assurance activities from the extended package show that the extended package for secure mobile messengers can be used to evaluate the security of existing products.

With the results from this work it is now possible to evaluate the security of other messengers. Further work should therefore be done in using this extended package to evaluate multiple messengers and compare their results.

A. Protection Profile

A.1. Introduction

A.1.1. Reference Identification

- PP Reference: Extended Package for Secure Mobile Messengers
- PP Version: 1.0
- PP Date: 01-Jun-2017

A.1.2. TOE Overview

This Extended Package defines requirements for the evaluation of Secure Mobile Messengers.

Such products are generally mobile software applications designed to conduct confidential and authenticated conversations over untrusted networks (i.e. the Internet).

Audio and video communications are outside of the scope of the current version of this PP, but are expected to be included in the scope of the next version.

A.1.3. TOE Use Cases

Secure Mobile Messengers perform tasks associated primarily with the following use case.

Sending and receiving messages

The application allows a user to communicate interactively and non-interactively with one or more other users over a secure channel. This includes exchanging text messages, media files and other types of information such as geological positions or contact information.

A.2. Conformance Claims

The Protection Profile for Application Software ([App PP]) defines the baseline Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) for application software products. This EP serves to extend the [App PP] baseline with additional SFRs and associated Assurance Activities specific to secure mobile messengers. Assurance Activities are the actions that the evaluator performs in order to determine a secure mobile messenger's compliance to the SFRs.

This EP conforms to Common Criteria [CC] for Information Technology Security Evaluation, Version 3.1, Revision 5. It is CC Part 2 extended and CC Part 3 conformant. In order to be conformant to this EP, the ST must include all components in this EP and the associated App PP that are: unconditional (which are always required), selection based (which are required when certain selections are chosen in the unconditional requirements) and may include optional and/or objective components that are desirable but not required for conformance.

In accordance with CC Part 1, dependencies are not included when they are addressed by other SFRs. The assurance activities provide adequate proof that any dependencies are also satisfied.

A.3. Security Problem Definition

The security problem is described in terms of the threats that a secure mobile messenger is expected to address, assumptions about the operational environment, and any organizational security policies that it is expected to enforce.

This Extended Package does not repeat the threats, assumptions, and organizational security policies identified in the App PP, though they all apply given the conformance and hence dependence of this EP on it. Together the threats, assumptions and organizational security policies of the App PP and those defined in this EP describe those addressed by an secure mobile messengers as the Target of Evaluation.

Notably, secure mobile messengers are particularly at risk from the Network Attack and Network Eavesdrop threats identified in the App PP. Attackers can send malicious messages directly to users, and the messenger clients will render or otherwise process this untrusted content.

A.3.1. Threats

The following threats are specific to secure mobile messengers and represent an addition to those identified in the [App PP].

- **T.UNSAFE_AUTHFACTOR_VERIFICATION** An attacker can take advantage of an unsafe method for performing verification of an authorization factor (e.g. SMS), resulting in exposure of the cryptographic material or user data.

- **T.WEAK_CRYPTOGRAPHY** An attacker may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort.
- **T.SECURITY_FUNCTIONALITY_FAILURE** A component of the mobile messenger (e.g. random number generator) may fail during start-up or during operations causing a compromise or failure in the security functionality of the messenger, leaving the messenger susceptible to attackers.

A.4. Security Objectives

The security objectives in this section were constructed to address threats identified in section A.3.1.

This Extended Package adds security objectives to those identified in the [App PP].

A.4.1. Security Objectives for the TOE

- **O.PROTECTED_COMMS** Addressed by: FTP_DIT_EXT.1
- **O.ACCESS** To address issues associated with the loss of confidentiality of user data in the event of loss of physical control of the device, conformant messengers implement mechanisms to restrict access to them. This includes automatic and manual access restrictions.
  Addressed by: FTA_SSL_EXT.1
- **O.AUTHENTICATION** To address issues associated with weak authentication mechanisms (e.g. identity theft, impersonation of other users), the messenger must ensure that it protects the authentication process by using strong authentication mechanisms (e.g. two-factor authentication).
  Addressed by: FIA_UAU_EXT.1, FIA_UAU.5, FIA_UAU.6, FIA_UAU.7

A.5. Security Functional Requirements

The Security Functional Requirements (SFRs) in this section are a formal instantiation of the security objectives form section A.4. The individual SFRs are specified in the sections below. SFRs in this section are mandatory SFRs that any conforming TOE must meet.

The SFRs included in this section are derived from Part 2 of the Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, with additional extended functional components.

The notations used in descriptions of the SFRs are as follows:

- Assignment: Indicated with *italicized text*;
- Refinement made by PP author: Indicated with **bold text** and ~~strikethroughs~~, if necessary;
- Selection: Indicated with underlined text;
- Assignment within a Selection: Indicated with *italicized and underlined text*;
- Iteration: Indicated by appending the iteration number in parenthesis, e.g., (1), (2), (3) and/or by adding a string starting with "/".

Extended SFRs are identified by having a label 'EXT' at the end of the SFR name.

A.5.1. TOE Access (FTA)

Session Locking (FTA_SSL)

TSF- and User-initiated locked state (FTA_SSL_EXT.1)	
FTA_SSL_EXT.1.1	The TSF shall transition to a locked state after a time interval of inactivity and a user initiated lock, and upon transitioning to the locked state, the TSF shall perform the following operations: *[assignment: actions performed upon transitioning to the locked state]*. **Assurance Activity:** The evaluator shall perform the following tests: Test 1: The evaluator unlocks the application and waits for the defined interval of inactivity. Then the evaluator shall verify that the application transitioned into the locked state and the defined actions were executed. Test 2: The evaluator should initiate a lock according to the procedures described in the documentation and shall verify that the defined actions were executed.
FTA_SSL_EXT.1.2	The TSF shall detect when *[assignment: range of acceptable values]* of unsuccessful unlocking attempts occur related to last successful unlocking by that user. When the defined number of unsuccessful unlocking attempts has been [selection: met, surpassed], the TSF shall perform *[assignment: action to execute]*. **Assurance Activity:** The evaluator shall try to unlock the application using procedures described in the documentation for the defined number of times and verify that the application performs the defined action.
FTA_SSL_EXT.1.3	The TSF shall provide only obscured feedback to the device's display to the user while the unlocking is in progress. **Assurance Activity:** The evaluator shall unlock the application and verify that the output is obscured.

A.5.2. Trusted Path/Channel (FTP)

Data in Transit (FTP_DIT)

Protection of Data in Transit (FTP_DIT_EXT.1)	
FTP_DIT_EXT.1.1	The application shall [selection: • encrypt all transmitted sensitive data with **[assignment: encryption scheme]**, • encrypt all transmitted data with **[assignment: encryption scheme]**] between itself and another trusted IT product. **Application Note:** Extended packages may override this requirement to provide for other protocols. Encryption is not required for data that is not sensitive. **Assurance Activity:** The evaluator shall exercise the application (attempting to transmit data) while capturing packets from the application. The evaluator shall review the packet capture and verify that no sensitive data is transmitted in the clear.

A.5.3. Identification and Authentication (FIA)

User Authentication (FIA_UAU)

Timing of Authentication (FIA_UAU_EXT.1)	
FIA_UAU_EXT.1.1	The TSF shall allow [selection: *[assignment: list of actions]*, no actions] on behalf of the user to be performed before the user is authenticated. **Assurance Activity:** The evaluator shall use the application before authentication and verify that the list of defined actions can be used.
FIA_UAU_EXT.1.2	The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user. **Assurance Activity:** The evaluator shall use the application before authentication and verify that no other than the defined actions can be used.

Multiple authentication mechanisms (FIA_UAU.5)	
FIA_UAU.5.1	The TSF shall provide *[assignment: list of multiple authentication mechanisms]* to support user authentication. **Assurance Activity:** The evaluator shall authenticate with the application and shall verify that all defined authentication mechanisms are supported.
FIA_UAU.5.2	The TSF shall authenticate any user's claimed identity according to *[assignment: the rules describing how the multiple authentication mechanisms provide authentication]*. **Assurance Activity:** The evaluator shall authenticate with the application using procedures described in the documentation and shall verify that the defined rules conform with the documentation.

Re-Authentication (FIA_UAU.6)	
FIA_UAU.6.1	The TSF shall re-authenticate the user under the conditions *[assignment: list of conditions under which re-authentication is required]*. **Assurance Activity:** The evaluator shall cause the defined conditions and shall verify that the user needs to be re-authenticated.

Protected Authentication Feedback (FIA_UAU.7)	
FIA_UAU.7.1	The TSF shall provide only [obscured feedback to the device's display] to the user while the authentication is in progress. **Assurance Activity:** The evaluator shall authenticate using each available authentication method and shall verify that the output is obscured.

B. Security Target for Telegram

B.1. Introduction (ASE_INT)

This section presents the following information required for a Common Criteria (CC) evaluation:

- Identification of the Security Target (ST) and the Target of Evaluation (TOE);
- TOE overview, which briefly describes the TOE;
- TOE description, which describes the TOE in more detail.

B.1.1. Reference Identification

Security Target Identification

- ST Title: Telegram Security Target
- ST Version: 1.0
- ST Author: Markus Müller
- ST Date: 01-Sep-2017

TOE Identification

- TOE Developer: Telegram Messenger LLP
- TOE Software Identification: Telegram Messenger
- TOE Versions: 4.0.0 (990)

B.1.2. TOE Overview

Usage and major security features of the TOE Telegram is a mobile messaging application for mobile operating systems (e.g. Android, iOS and Windows Mobile) with a focus on security. Users can send messages, photos, videos and any other files to other users. Users are identified by their phone numbers and optional usernames. Messages to other users are end-to-end encrypted. They can have an additional self-destruction timer that deletes the content after a defined amount of time. The access to the application can be controlled with an additional passcode.

TOE Type Application for mobile devices

Required non-TOE hardware/software/firmware

- Software Requirements: The TOE runs on Android version 4.4.
- Hardware Requirements: The TOE imposes no hardware requirements beyond Android operating system requirements.

B.1.3. TOE Description

After a brief overview of the Telegram product, this section describes the Telegram client application, which is the Target of Evaluation (TOE). The description covers TOE architecture, logical boundaries, and physical boundaries.

TOE Architecture

The Telegram Messenger consists of the Telegram client application and the Telegram service. The Telegram client is a mobile application that connects to the Telegram service for providing its features.

Messaging Users can write end-to-end encrypted messages to other users (Secret chats), so the content of the message cannot be read by anyone except of the communicating parties. Other users are identified by their phone numbers (taken from the users address book) or by usernames. Users can choose an optional username for easier identification.

Self-destructing messages Messages in Secret Chats can be ordered to self-destruct. As soon as such a message is read and a defined amount of time has passed, both devices participating in the chat are instructed to delete the message.

Media and other files In addition to sending text messages, users can exchange photos, videos and other files. When these files are sent via secret chats, the files are first encrypted and then put on a server of the Telegram service. The encryption key and the location are then sent to the other user, so he can download and decrypt the file.

Authentication The Telegram service has a directory of users using Telegram. Telegram uses phone numbers as unique identifiers. For using the Telegram service, a user first needs to register with the service. To find other users, the Telegram client synchronizes the users contacts (i.e. the address book) with the Telegram service. As soon as a users contact signs up for Telegram, the user will be notified of this event.

Registration For registration, the user requests a new account with his phone number. The Telegram service then sends an SMS with a random number to the user's phone. When the user enters the correct number, he is then registered as a user with his phone number.

Second Authentication Factor For improving the security, a user can configure a second factor (password) for authentication, which is checked during login. The user can also set up a recovery email address. This address will be used to receive a password recovery code, with which the user can reset his password.

Account deletion The user can request the deletion of his account, which includes all associated data (e.g. personal contacts). The Telegram service deletes accounts that were not used (i.e. no login) for at least 6 months. The user can define the exact period after which his inactive account will self-destruct.

Physical Boundary

The TOE consists of Telegram client application and configuration settings as defined in the Telegram installation package for Android. The Telegram application is a client that only communicates with the Telegram service. The Telegram service and any functions not specified in this security target are outside the scope of the TOE.

The TOE guidance documentation that is considered to be part of the TOE can be found listed in this document in chapter C.

Logical Boundary (Security Functions provided by the TOE)

The TOE provides the security functionality required by [SMM EP]. Each of the security features identified consists of several security functions, as described in more detail in the subsections below.

Cryptographic Support The TOE provides cryptography in support of other Telegram security functionality.

The cryptographic services provided by the TOE are described in table B.1 below.

Cryptographic Method	Use within the TOE
AES	Used to encrypt messages between users
	Used for encrypting media and files
RNG	Used for random number generation
	Used in authentication process
	Used for message padding
SHS (SHA-1)	Used for visualization of user's key fingerprints
	Used for key derivation for messages
MD5	Used for user's key fingerprints

Table B.1.: Cryptographic services for Telegram security functionality

User Data Protection The TOE ensures that all access to platform resources (e.g. address book or camera) is restricted to the defined resources.

The TOE leverages platform provided services for encrypting sensitive application data.

Identification and Authentication The TOE performs registration and authentication for the user of the TOE with the Telegram service. The user can terminate active sessions. Once a session has been terminated, the TOE requires the user to re-authenticate to establish a new session.

Security Management The TOE provides capabilities to manage its security functions. All TOE administration occurs through the settings menu in the application. The TOE provides the ability to securely manage:

- all identification and authentication;
- the passcode protection of the TOE;
- active sessions with the Telegram service.

Privacy The TOE lets the user select which personally identifiable information are visible to other users.

Protection of the TSF The TOE employs several mechanisms to ensure that it is secure on the host platform: The TOE

- uses only documented platform APIs;
- never allocates memory with both write and execute permissions;
- is designed to operate in an environment in which the following security techniques are in effect: stack-based buffer overflow protection, Data execution prevention (DEP), Address space layout randomization (ASLR).

The TOE relies on the platform for software distribution, installation and updates.

TOE Access For restricting access to the TOE while it is authenticated for the Telegram service, the user can configure a passcode. After a defined timeout or on user request the TOE can be transitioned into a locked state.

Trusted Path/Channels The TOE secures its connection to the Telegram service with a custom encryption protocol. Messages and data exchanges between users are also secured with a custom encryption protocol (end-to-end encryption).

B.2. Conformance Claims (ASE_CCL)

This Security Target is CC Part 2 extended and CC Part 3 conformant. It claims conformance to the following Protection Profile:

- Application Software Protection Profile (App PP). Version 1.2 as of 2016; strict conformance.
- Secure Mobile Messenger Extended Package (SMM EP). Version 1.0 as of 2017; strict conformance.

Common Criteria [CC] version 3.1 revision 5 is the basis for this conformance claim.

B.3. Security Problem Definition (ASE_SPD)

B.3.1. Threats

This section identifies the threats against the TOE. These threats have been taken from the [App PP] and [SMM EP].

- **T.NETWORK_ATTACK** An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may engage in communications with the application software or alter communications between the application software and other endpoints in order to compromise it.

- **T.NETWORK_EAVESDROP** An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may monitor and gain access to data exchanged between the application and other endpoints.
- **T.LOCAL_ATTACK** An attacker can act through unprivileged software on the same computing platform on which the application executes. Attackers may provide maliciously formatted input to the application in the form of files or other local communications.
- **T.PHYSICAL_ACCESS** An attacker may try to access sensitive data at rest.
- **T.UNSAFE_AUTHFACTOR_VERIFICATION** An attacker can take advantage of an unsafe method for performing verification of an authorization factor (e.g. SMS), resulting in exposure of the cryptographic material or user data.
- **T.SECURITY_FUNCTIONALITY_FAILURE** A component of the mobile messenger (e.g. random number generator) may fail during start-up or during operations causing a compromise or failure in the security functionality of the messenger, leaving it vulnerable to attackers.

B.3.2. Organisational security policies (OSPs)

Neither [App PP] nor [SMM EP] define organizational security policies.

B.3.3. Assumptions

The following assumptions have been taken from the [App PP]. The [SMM EP] defines no further assumptions.

A.PLATFORM The TOE relies upon a trustworthy computing platform for its execution. This includes the underlying platform and whatever runtime environment it provides to the TOE. **A.PROPER_USER** The user of the application software is not willfully negligent or hostile, and uses the software in compliance with the applied enterprise security policy. **T.PROPER_ADMIN** The administrator of the application software is not careless, willfully negligent or hostile, and administers the software within compliance of the applied enterprise security policy.

B.4. Security Objectives (ASE_OBJ)

This section identifies the security objectives of the TOE and its supporting environment. The security objectives identify the responsibilities of the TOE and its environment in meeting the security needs.

The complete security problem definition may be found in [App PP] and [SMM EP] and this section reproduces only the corresponding security objectives for for reader convenience. The [App PP] and [SMM EP] offer additional information about the identified security objectives, but that has not been reproduced here and both the [App PP] and the [SMM EP] should be consulted if there is interest in that material.

B.4.1. Security Objectives for the TOE

This section identifies the security objectives of the TOE as defined by [App PP] and [SMM EP].

- **O.INTEGRITY** Conformant TOEs ensure the integrity of their installation and update packages, and also leverage execution environment-based mitigations. Software is seldom if ever shipped without errors, and the ability to deploy patches and updates to fielded software with integrity is critical to enterprise network security. Processor manufacturers, compiler developers, execution environment vendors, and operating system vendors have developed execution environment-based mitigations that increase the cost to attackers by adding complexity to the task of compromising systems. Application software can often take advantage of these mechanisms by using APIs provided by the runtime environment or by enabling the mechanism through compiler or linker options.
- **O.QUALITY** To ensure quality of implementation, conformant TOEs leverage services and APIs provided by the runtime environment rather than implementing their own versions of these services and APIs. This is especially important for cryptographic services and other complex operations such as file and media parsing. Leveraging this platform behavior relies upon using only documented and supported APIs.
- **O.MANAGEMENT**To facilitate management by users and the enterprise, conformant TOEs provide consistent and supported interfaces for their security-relevant configuration and maintenance. This includes the deployment of applications and application updates through the use of platform-supported deployment mechanisms and formats, as well as providing mechanisms for configuration. This also includes providing control to the user regarding disclosure of any PII.
- **O.PROTECTED_STORAGE** To address the issue of loss of confidentiality of user data in the event of loss of physical control of the storage medium, conformant TOEs will use data-at-rest protection. This involves encrypting data and keys stored by the TOE in order to prevent unauthorized access to this data. This also includes unnecessary network communications whose consequence may be the loss of data.
- **O.PROTECTED_COMMS** To address both passive (eavesdropping) and active (packet modification) network attack threats, conformant TOEs will use a trusted

channel for sensitive data. Sensitive data includes cryptographic keys, passwords, and any other data specific to the application that should not be exposed outside of the application.

- **O.ACCESS** To address issues associated with the loss of confidentiality of user data in the event of loss of physical control of the device, conformant messengers implement mechanisms to restrict access to them. This includes automatic and manual access restrictions.
- **O.AUTHENTICATION** To address issues associated with weak authentication mechanisms (e.g. identity theft, impersonation of other users), the messenger must ensure that it protects the authentication process by using strong authentication mechanisms (e.g. two-factor authentication).

B.4.2. Security Objectives for the Operational Environment

The TOE's operating environment must satisfy the following objectives, as defined by [App PP] and [SMM EP].

- **OE.PLATFORM** The TOE relies upon a trustworthy computing platform for its execution. This includes the underlying operating system and any discrete execution environment provided to the TOE.
- **OE.PROPER_USER** The user of the application software is not willfully negligent or hostile, and uses the software within compliance of the applied enterprise security policy.
- **OE.PROPER_ADMIN** The administrator of the application software is not careless, willfully negligent or hostile, and administers the software within compliance of the applied enterprise security policy.

B.4.3. Security Problem Definition Rationale

The assumptions, threats, OSPs, and objectives that are defined in this ST represent the assumptions, threats, OSPs, and objectives that are specified in the Protection Profile to which the TOE claims conformance. The associated mappings of assumptions to environmental objectives, SFRs to TOE objectives, and OSPs and objectives to threats are therefore identical to the mappings that are specified in the claimed protection profiles.

B.5. Extended Components Definition (ASE_ECD)

B.5.1. Extended Security Functional Requirements

The extended Security Functional Requirements that are claimed in this ST are taken directly from the PP to which the ST and TOE claim conformance. These extended components are formally defined in the PP in which their usage is required.

B.5.2. Extended Security Assurance Requirements

The extended Security Assurance Requirement that is claimed in this ST is taken directly from the PP to which the ST and TOE claim conformance. This extended component is formally defined in the PP in which its usage is required.

B.6. Security requirements (ASE_REQ)

The notations used in descriptions of the SFRs are as follows:

- Assignment: Indicated with *italicized text*;
- Refinement made by PP author: Indicated with **bold text** and ~~strikethroughs~~, if necessary;
- Selection: Indicated with underlined text;
- Assignment within a Selection: Indicated with *italicized and underlined text*;
- Iteration: Indicated by appending the iteration number in parenthesis, e.g., (1), (2), (3) and/or by adding a string starting with "/".

Extended SFRs are identified by having a label ’EXT’ at the end of the SFR name. Table B.2 lists the SFRs claimed by the TOE.

B.6.1. Cryptographic Support (FCS)

Random Bit Generation (FCS_RBG)

Random Bit Generation (FCS_RBG_EXT.1)	
FCS_RBG_EXT.1.1	The application shall invoke platform-provided DRBG functionality for its cryptographic operations.

Storage of Credentials (FCS_STO)

Storage of Credentials (FCS_STO_EXT.1)	
FCS_STO_EXT.1.1	The application shall [not store any credentials] to non-volatile memory.

Functional Class	**Functional Components**
Cryptographic Support (FCS)	FCS_RBG_EXT.1 Random Bit Generation
	FCS_STO_EXT.1 Storage of Credentials
User Data Protection (FDP)	FDP_DEC_EXT.1 Access to Platform Resources
	FDP_NET_EXT.1 Network Communications
	FDP_DAR_EXT.1 Encryption Of Sensitive Application Data
Security Management (FMT)	FMT_MEC_EXT.1 Supported Configuration Mechanism
	FMT_CFG_EXT.1 Secure by Default Configuration
	FMT_SMF.1 Specification of Management Functions
Privacy (FPR)	FPR_ANO_EXT.1 User Consent for Transmission of PII
Protection of the TSF (FPT)	FPT_API_EXT.1 Use of Supported Services and APIs
	FPT_AEX_EXT.1 Anti-Exploitation Capabilities
	FPT_TUD_EXT.1 Integrity for Installation and Update
	FPT_LIB_EXT.1 Use of Third Party Libraries
TOE Access (FTA)	FTA_SSL_EXT.1 TSF- and User-initiated locked state
Trusted Path/Channel (FTP)	FTP_DIT_EXT.1 Protection of Data in Transit
Identification and Authentication (FIA)	FIA_UAU_EXT.1 Timing of Authentication
	FIA_UAU.5 Multiple authentication mechanisms
	FIA_UAU.6 Re-Authentication
	FIA_UAU.7 Protected Authentication Feedback

Table B.2.: SFRs claimed by Telegram

B.6.2. User Data Protection (FDP)

Platform Resources (FDP_DEC)

Access to Platform Resources (FDP_DEC_EXT.1)	
FDP_DEC_EXT.1.1	The application shall restrict its access to network connectivity, camera, microphone and location services.
FDP_DEC_EXT.1.2	The application shall restrict its access to address book.

Network Communications (FDP_NET)

Network Communications (FDP_NET_EXT.1)	
FDP_NET_EXT.1.1	The application shall restrict network communication to *user authentication and exchanging messages*.

Application Data (FDP_DAR)

Encryption Of Sensitive Application Data (FDP_DAR_EXT.1)	
FDP_DAR_EXT.1.1	The application shall not store any sensitive data in non-volatile memory.

B.6.3. Security Management (FMT)

Configuration Mechanism (FMT_MEC)

Supported Configuration Mechanism (FMT_MEC_EXT.1)	
FMT_MEC_EXT.1.1	The application shall invoke the mechanisms recommended by the platform vendor for storing and setting configuration options.

Default Configuration (FMT_CFG)

Secure by Default Configuration (FMT_CFG_EXT.1)	
FMT_CFG_EXT.1.1	The application shall provide only enough functionality to set new credentials when configured with default credentials or no credentials.

Specification of Management Functions (FMT_SMF)

Specification of Management Functions (FMT_SMF.1)	
FMT_SMF.1.1	The TSF shall be capable of performing the following management functions [• *managing two-step authentication,* • *managing active sessions,* • *define time-out interval for automatic transition into the locked state,* • *change passcode for unlocking the TOE*].

B.6.4. Privacy (FPR)

Personally Identifiable Information (FPR_ANO)

User Consent for Transmission of PII (FPR_ANO_EXT.1)	
FPR_ANO_EXT.1.1	The application shall require user approval before executing *setting or changing the user's name*.

B.6.5. Protection of the TSF (FPT)

Supported Services and APIs (FPT_API)

Use of Supported Services and APIs (FPT_API_EXT.1)	
FPT_API_EXT.1.1	The application shall use only documented platform APIs.

Anti-Exploitation (FPT_AEX)

Anti-Exploitation Capabilities (FPT_AEX_EXT.1)	
FPT_AEX_EXT.1.1	The application shall not request to map memory at an explicit address except for *no exception*.
FPT_AEX_EXT.1.2	The application shall not allocate any memory region with both write and execute permissions.
FPT_AEX_EXT.1.3	The application shall be compatible with security features provided by the platform vendor.
FPT_AEX_EXT.1.4	The application shall not write user-modifiable files to directories that contain executable files unless explicitly directed by the user to do so.
FPT_AEX_EXT.1.5	The application shall be compiled with stack-based buffer overflow protection enabled.

Installation and Update (FPT_TUD)

Integrity for Installation and Update (FPT_TUD_EXT.1)	
FPT_TUD_EXT.1.1	The application shall leverage the platform to check for updates and patches to the application software.
FPT_TUD_EXT.1.2	The application shall be distributed using the format of the platform-supported package manager.
FPT_TUD_EXT.1.3	The application shall be packaged such that its removal results in the deletion of all traces of the application, with the exception of configuration settings, output files, and audit/log events.
FPT_TUD_EXT.1.4	The application shall not download, modify, replace or update its own binary code.
FPT_TUD_EXT.1.5	The application shall leverage the platform to query the current version of the application software.
FPT_TUD_EXT.1.6	The application installation package and its updates shall be digitally signed such that its platform can cryptographically verify them prior to installation.

Third Party Libraries (FPT_LIB)

Use of Third Party Libraries (FPT_LIB_EXT.1)	
FPT_LIB_EXT.1.1	The application shall be packaged with only *[* • boringssl in version 3.2.6 • breakpad in version 3.2.6 • ffmpeg in version 3.4.2 • libjpeg in version 3.2.6 • libwebp in version 3.2.6 • libyuv in version 3.10.1 • opus in version 3.2.6 • sqlite in version 3.9.0 *]*.

B.6.6. TOE Access (FTA)

Session Locking (FTA_SSL)

TSF- and User-initiated locked state (FTA_SSL_EXT.1)	
FTA_SSL_EXT.1.1	The TSF shall transition to a locked state after a time interval of inactivity and a user initiated lock, and upon transitioning to the locked state, the TSF shall perform the following operations: *[no action]*.
FTA_SSL_EXT.1.2	The TSF shall detect when *[10]* of unsuccessful unlocking attempts occur related to last successful unlocking by that user. When the defined number of unsuccessful unlocking attempts has been [met], the TSF shall perform *[no action]*.
FTA_SSL_EXT.1.3	The TSF shall provide only obscured feedback to the device's display to the user while the unlocking is in progress.

B.6.7. Trusted Path/Channel (FTP)

Data in Transit (FTP_DIT)

Protection of Data in Transit (FTP_DIT_EXT.1)	
FTP_DIT_EXT.1.1	The application shall encrypt all transmitted sensitive data with *a custom encryption scheme* between itself and another trusted IT product.

B.6.8. Identification and Authentication (FIA)

User Authentication (FIA_UAU)

Timing of Authentication (FIA_UAU_EXT.1)	
FIA_UAU_EXT.1.1	The TSF shall allow *registration with the service* on behalf of the user to be performed before the user is authenticated.
FIA_UAU_EXT.1.2	The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Multiple authentication mechanisms (FIA_UAU.5)	
FIA_UAU.5.1	The TSF shall provide *SMS token with additional user-defined password* to support user authentication.
FIA_UAU.5.2	The TSF shall authenticate any user's claimed identity according to *the verification of the number received via SMS and a user-defined password*.

Re-Authentication (FIA_UAU.6)	
FIA_UAU.6.1	The TSF shall re-authenticate the user under the conditions *session termination, session expiration*.

Protected Authentication Feedback (FIA_UAU.7)	
FIA_UAU.7.1	The TSF shall provide only *obscured feedback to the device's display* to the user while the authentication is in progress.

B.6.9. Statement of Security Functional Requirements Consistency

The Security Functional Requirements included in the ST represent all required SFRs specified in the claimed PP as well as a subset of the optional SFRs. All hierarchical relationships, dependencies, and unfulfilled dependency rationales in the ST are considered to be identical to those that are defined in the claimed PP.

B.7. Security Assurance Requirements

Because the ST and TOE claim exact conformance to [App PP] and [SMM EP], the Security Assurance Requirements (SARs) that are claimed are identical to those defined in the claimed PP and have not been reproduced.

B.8. TOE summary specification

This section describes the security functions of the Telegram messenger.

B.8.1. Cryptographic services

Telegram uses the platform provided random number generator (/dev/urandom) as an entropy source. It stores no credentials on the device.

This security function covers the SFRs of: FCS_RBG_EXT.1 (Random Bit Generation), FCS_STO_EXT.1 (Storage of Credentials)

B.8.2. User data protection

Use of Encryption

Telegram relies on the underlying platform (Android) to provide encryption for data at rest.

The following table lists the data at rest that is secured by the Operational Environment:

- Messages sent and received
- Media and files sent and received

Apart from that Telegram stores no sensitive data.

This security function covers the SFRs of: FDP_DAR_EXT.1 (Encryption Of Sensitive Application Data)

Platform Resources and Networking

Telegram requires network access to authenticate users and exchange messages. It relies on its underlying platform to provide network connectivity.

Telegram needs access to camera, microphone and location services for sending pictures, voice messages and positions to other users. For finding other users, the product also needs access to the address book.

This security function covers the SFRs of: FDP_DEC_EXT.1 (Access to Platform Resources), FDP_NET_EXT.1 (Network Communications)

B.8.3. Identification and authentication

A user needs an account with the Telegram service to use it. To get an account a user can register with the service before he is authenticated.

Telegram can authenticate users based on their phone numbers and an additional password. The logons start with the user providing a phone number, to which a random number is sent. When the received number is entered correctly, the user is asked for his optional additional password.

A user can manage his active sessions (see section B.8.4). If the user or the Telegram service terminates an active session, the user has to re-authenticate.

When a user enters their authentication data, the TOE does not echo any characters as they are entered and as such the user password is obscured.

This security function covers the SFRs of: FIA_UAU_EXT.1 (Timing of Authentication), FIA_UAU.5 (Multiple authentication mechanisms), FIA_UAU.6 (Re-Authentication), FIA_UAU.7 (Protected Authentication Feedback)

B.8.4. Security management

Telegram provides the user with the capability to administer the security functions described in this security target. The user can perform these functions via the settings menu in the application, which are saved in an XML file at location $/data/data/package/shared_prefs/$. The specific management capabilities available in Telegram include:

1. Set, change and remove passcode lock
2. Activate, change and deactivate two-step verification
3. List and terminate active sessions
4. Change self-destruction time for the account

When Telegram is started for the first time, the user can only register a new account or authenticate with an existing one.

This security function covers the SFRs of: FMT_CFG_EXT.1 (Secure by Default Configuration), FMT_SMF.1 (Specification of Management Functions), Supported Configuration Mechanism (FMT_MEC_EXT.1)

B.8.5. Privacy

Telegram uses the following personally identifiable information:

- name
- profile pictures

Telegram requires the user to set a first name and an optional last name. Using profile pictures is optional too, so the user can decide if he wants to use one.

This security function covers the SFRs of: FPR_ANO_EXT.1 (User Consent for Transmission of PII)

B.8.6. Protection of the TSF

Telegram leverages the Google Android platform for protecting its security functions.The application is written in Java and uses a C++ library (tgnet) for the custom networking protocol. It uses only documented APIs.

Telegram updates

Telegram uses the App Store of the platform (Google Play Store) to identify the current version of the Telegram application and for updating.

Updates to the Telegram application are delivered as Android Package Kit files (.apk files) which are signed by Telegram. The certificates are checked by the Google Play store before installing the update.

This security function covers the SFRs of: FPT_TUD_EXT.1 (Integrity for Installation and Update)

Third Party Libraries

The TOE uses the following third party libraries:

- Google Play Services (com.google.android.gms) for in version 10.2.0 for Google Cloud Messaging, Google Maps and Mobile Vision
- Android Support Libraries (com.android.support) in version 25.3.0 for providing newer Android features on earlier versions of Android
- HockeyApp (net.hockeyapp.android) in version 4.1.2 for crash reports and feedback
- Java MP4 Parser (com.googlecode.mp4parser) in version 1.0.6 for parsing and writing ISO 14496 based files (e.g. MP4)
- Stripe (com.stripe) in version 2.0.2 for payments
- boringssl in version 3.2.6
- breakpad in version 3.2.6
- ffmpeg in version 3.4.2
- libjpeg in version 3.2.6
- libwebp in version 3.2.6
- libyuv in version 3.10.1
- opus in version 3.2.6
- sqlite in version 3.9.0

This security function covers the SFRs of: FPT_LIB_EXT.1 (Use of Third Party Libraries)

Supported Services and APIs

The TOE leverages the following platform provided Application Programming Interfaces (APIs):

- android.accounts
- android.animation
- android.annotation
- android.app
- android.bluetooth
- android.content
- android.database
- android.graphics
- android.hardware
- android.location
- android.media
- android.net
- android.opengl
- android.os
- android.provider
- android.service
- android.support
- android.system
- android.telephony
- android.text
- android.util
- android.view
- android.webkit
- android.widget
- com.android.internal.telephony
- org.json
- org.xmlpull.v1
- org.xml.sax

This security function covers the SFRs of: FPT_API_EXT.1 (Use of Supported Services and APIs)

Plaintext Key Storage

Telegram writes no keys to persistent storage. For every connections it generates new random secrets and does a Diffie-Hellman handshake with the other parties (i.e. the

Telegram service and other users).

This security function covers the SFRs of: FPT_KST_EXT.1 (Plaintext Key Storage)

B.8.7. TOE Access

Telegram provides the ability for a user to lock the application after a user-defined inactivity timeout.

After the Telegram application was locked, in order to unlock the application, the user opens it and is presented with an authentication dialog. The user must then re-enter his passcode, after which the user's conversations and contacts will be visible again.

The Telegram application maintains a count of consecutive failed unlocking attempts by users from their last successful authentication. When the number of consecutive failed unlocking attempts is larger than 5, the application will restrict further attempts for one minute.

This security function covers the SFRs of: FTA_SSL_EXT.1 (TSF- and User-initiated locked state)

B.8.8. Trusted path/channels

Telegram provides trusted network channels to register with the Telegram service and communicate with other users.

It uses a custom cryptographic protocol for exchanging messages and files with other users.

This security function covers the SFRs of: FTP_DIT_EXT.1 (Protection of Data in Transit)

C. Guidance Documents for Telegram

This chapter provides the required documentation for Telegram.

C.1. Installing and Updating

See the Google Play store documentation for checking application versions as well as installing and updating Telegram.

C.2. Management

This section describes the following management functions:

- managing two-step authentication,
- managing active sessions,
- define time-out interval for automatic transition into the locked state,
- changing passcode for unlocking Telegram.

C.2.1. 2FA

Two step verification allows the user to set up a password that will be required every time he logs into his account from a new device - in addition to the code he gets in the SMS.

If the user forgets this password, he will not be able to access his messages from other devices anymore. It is therefore recommended to set up a recovery e-mail or at least a hint for your password.

To turn it on, the user needs to go to *Settings - Privacy and Security*.

C.2.2. Active Sessions

Telegram has Active Sessions to the Privacy and Security settings. This screen shows all logged in devices with IP info. A user can also close any sessions that are outdated or suspicious.

To terminate Telegram sessions go to *Settings - Privacy and Security - Active Sessions*.

C.2.3. Passcode

In order to set up a passcode, go to *Telegram Settings - Privacy & Security - Passcode*. Once a passcode is set up, a lock icon appears in the chats list. The user needs to tap on it to lock the app and the passcode will be required next time the Telegram application is opened. When the app is locked, text and sender's name are hidden in notifications.

A user can also enable auto-lock in Passcode Settings, if he does not want to lock Telegram manually. At the moment the smallest available setting is 1 minute.

D. Management Report

As a user of mobile messengers, it is rather difficult to verify the security claims the messenger's creators made. Depending on the messenger, so far there are no or only limited independent analyses on the state of security. Thus, the aim of this work is to describe requirements for secure mobile messengers and evaluate the Telegram messenger according to these requirements.

This research provides commercial potential in a way that it enables the author of this work to provide consulting for Common Criteria certification, mobile messaging and security to the developers of Telegram and similar messaging products.

D.1. Project Planning

The work can be split into three different parts that are based on each other.

First, the author needs to define requirements for a secure mobile messenger according to the Common Criteria. This requirements are described in a standardized form as a Protection Profile (PP). Besides the security requirements it also defines threats and assumptions for the evaluated product as well as security function requirements - detailed descriptions of how security should be archived.

The next step is to create a Security Target (ST) for the Telegram messenger. This is a description of how Telegram provides the security requirements defined in the PP.

Then, the author needs to evaluate whether the ST complies with the PP and whether the implemented version (i.e. the Telegram messenger) complies with the ST. The findings of this evaluation are then summarized and based on the this the author decides whether Telegram fulfills the security requirements of the PP.

Table D.1 provides a work breakdown structure for this project and figure D.1 shows the project tasks as a Gantt chart.

Task	**Start**	**End**	**Duration**
Protection Profile	01.01.	31.01.	30
Security Target	01.02.	15.03.	42
Evaluation	16.03.	15.05.	60

Table D.1.: WBS for the project

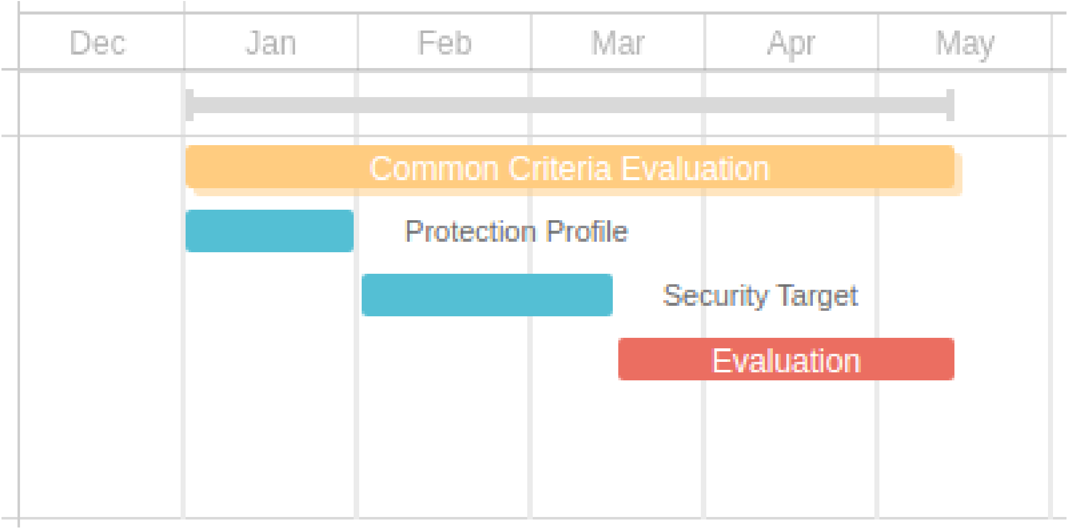

Figure D.1.: Gantt chart for the project

Item	**Cost per Item**	**Number of Items**	**Sum**
Smartphone	∼ 300 Euros	2	600 Euros
Computer	∼ 500 Euros	1	500 Euros
Electricity	∼ 30 Cent kw/h	250	75 Euros
Internet Connectivity	∼ 25 Euro / month	5	125 Euros

Table D.2.: Costs for hardware and utilities

D.2. Financial Management

The project outlined in the previous section does not require specialized resources.

The material required is limited to two devices capable of running the Telegram messenger (i.e. a smartphone), a computer for examining the application and writing the report as well as electricity and an internet connection. Table D.2 gives an overview over the estimated costs.

The more expensive part of this work is the labor costs, which involve the salary of the academic advisor and the author of this work. The estimated labor costs are listed in table D.3.

Person	Cost per hour	Hours	Sum
Academic Advisor	50 Euros	50	2500 Euros
Student	-	750	0 Euros

Table D.3.: Labor costs

D.3. Competitive Analysis

The following sections provide a SWOT and a PEST analysis for providing consulting services for Common Criteria evaluation.

D.3.1. SWOT

A SWOT analysis lists strengths, weaknesses, opportunities and threats for a product or service.

Strengths

- The author already did a certification for the Telegram messenger

Weaknesses

- The author has no further experience with certifications yet
- The experience is limited to one messenger

Opportunities

- Based on the experience with writing PPs/STs and evaluating products the author can move into other areas for certification (other products)
- The author can evaluate more messengers

Threats

- Other companies offering consulting for certification can move into this area
- Mobile messengers become irrelevant by a new technology
- Telegram goes bankrupt

D.3.2. PEST

A PEST analysis provides an overview over political, economical, social and technological factors that might influence the success of a product or service.

Political factors

- The government forbids cryptography, in which case the use of secure mobile messengers will be also forbidden.
- The government regulates the internet and blocks access to Telegram, by which the product looked at in this work becomes irrelevant.

Economic factors

- Other messengers (e.g. Signal or Threema) can grow larger and take Telegrams market share (network effects).
- Telegram goes bankrupt because it has no viable business model and its current sponsor (Pavel Durov) provides no further financing.

Social factors

- Mobile messengers become less popular in general and therefore also Telegram becomes irrelevant.
- Security issues with Telegram or other messengers move people away from it; they replace it with other secure communication technologies.
- Telegram loses user's trust by e.g. selling personal information about the users, so the users move away to alternatives.

Technological

- New technology replaces mobile messengers or smartphones.
- New technology (e.g. quantum computing) breaks used cryptographic methods and makes the messenger insecure and therefore useless.

E. Social Responsibility

The purpose of this section is to provide the reader with a solid understanding of the social responsibility relevant in the development context of mobile messengers. General but also more specific issues in this regard will be discussed.

Researchers agree that working with the computer affects an individual's overall health, especially their vision and posture. This chapter of this thesis focuses on the rules and norms for working on the computer which support favorable conditions regarding the individual's health in the software development process. In order to avoid negative health effects, this chapter will now analyze and identify measures for a healthy and safe work conditions. These include general industrial safety measures for the protection of dangerous and harmful factors, emergency behaviors and environmental protection.

E.1. Occupational Safety

In the development process of mobile messengers, several safety efforts for the software developer need to be respected. Any organization needs to ensure that their employees work in an environment free from recognized hazards, such as exposure to toxic chemicals, excessive noise levels, mechanical dangers, heat or cold stress, or unsanitary conditions.

E.1.1. Identification and analysis of workplace hazards, which the research object can create for people

Software development by nature is done in front of a computer. This inevitably results into a primarily sedentary workday. Sedentary lifestyles carry at least twice the risk of a serious disease and premature death and end up being a risk factor on par with the relative risk of hypertension and hyperlipidemia, close to the dangers of smoking (Biddle, Fox, & Boutcher, 2003). While seated, the body is still and circulation slows, reducing the supply of oxygen and nutrients to the muscles. This scenario, coupled with poor posture, can produce several musculoskeletal disorders (MSD) which eventually manifest with pain, tingling, discomfort, numbness and swelling in the joints and muscles. These negative health effects are mostly temporary, nonetheless, others can be permanent.

Accompanied by the work in front of a computer is the fact that the developer is primarily looking at a computer screen for most of his work. There is no evidence yet, that looking at a computer screen degrades vision permanently. However, short-term symptoms are common: People who use computers daily at work or at home could suffer from computer vision syndrome, which leaves them vulnerable to problems like dry eye, eyestrain, neck and backaches, light sensitivity and fatigue. Many of these symptoms result from poor workstation configuration and improper work habits. Although most of these symptoms cease once the individual is off the computer, some people will continue to experience visual problems, such as blurred distance vision.

When examining the literature, there seems to be an association between computer work and the so called carpal tunnel syndrome. The carpal tunnel syndrome is a medical condition due to the compression of the median nerve passing through the carpal tunnel, located in the wrist. It is regarded as the most frequent compression neuropathy. It has been a matter of discussion among researchers, whether computer work could be a risk factor for the development of the carpal tunnel syndrome.

Generally, it is accepted that exposure to hand-arm vibrations and exposure to a combination of repetitive hand use and the use of hand force serve as causal agents (Palmer et al., 2007). To test, whether the work on a keyboard and mouse supports the occurrence of the syndrome, a recent study from 2008 conducted by Thomsen et al. was consulted. The researchers conducted a systematic review of studies of computer work and the carpal tunnel syndrome. Supplementary, longitudinal studies of low force, repetitive work and carpal tunnel syndromes as well as studies of possible pathophysiological mechanisms were evaluated. Because of insufficient quality, bias, lack of consistency and statistical power, the evidence for a positive relationship between an exposure to computer work and the outcome of a carpal tunnel syndrome cannot be confirmed (Thompsen et al., 2008). Further research is needed if the condition can be recognized as an injury because of computer work.

According to the American Physical Therapy Association's Occupational Health Special Interest Group, text messaging and other handheld-based activities make IT professionals more vulnerable to developing symptoms ranging from hand throbbing and swelling to tendonitis. When text messaging, people tend to tense their shoulders and upper arms, which cuts down circulation to the forearm at the time when the consistent movements of the thumb and fingers require increased blood flow. Also, overuse can inflame underlying arthritis, further increasing the risk of injury.

Further workspace hazards concern the electronics the developer is working with. All regular electronic devices in a IT office need a power supply through an alternating voltage of 220V. When fully functioning, the risk of an electric malfunctioning can be classified as low, since manufacturers already take many precautions to fabricate safe products. However, there is still a risk of an electric shock. The computer can be energized through technical failures, damages or isolation breakdowns. In all those cases, there is a high risk of electric shock or other risk caused by electricity (i.e. electric burns) for the individual. Direct contact with live parts during a PC repair or the contact with dead parts or casings appearing under voltage can lead to an electric shock. Depending on the age and conditions of the devices, the danger of an electric shock can be increased or decreased.

Other workplace hazards concern the direct environment the developer finds himself to be in. Key parameters for the individuals' well-being concern the indoor climate such as air temperature, relative humidity, and air velocity. An optimal combination of temperature, relative humidity and air velocity are known to create a comfortable working environment (Witterseh, Wyon, & Clausen, 2004).

E.1.2. Identification and analysis of workplace hazards, which may influence a researcher during the research process

During the research process, work-related stress might motivate people in manageable doses, but can grind down overall health over time. Many lifestyle diseases such as chronic fatigue, depression and obesity are results of workplace-induced habits such as high stress levels, low physical activity, and disturbed biological clocks (Sharma & Majumdar, 2009).

Stress can lower your immune defenses, increase the risk of heart disease and bring on anxiety, depression and difficulty sleeping. If a major portion of the worktime every day is consumed in front of a computer, an overstimulation of the brain can occur. Without implementing a consistent exercise regimen to boost brain endorphins or allotting the proper downtime for mental relaxation, overworked IT professionals leave themselves vulnerable to increased stress.

In times of stress, the brain releases adrenaline and other hormones to heighten senses and boost strength. If the stress amount turns into a chronic stress, the workers immune and cardiovascular systems can be harmed. Further, an increased vulnerability to heart disease, depression, exhaustion, sleep deprivation and overall malaise might occur.

E.1.3. Protection methods to mitigate the potential damage

Attempts to ensure safe working conditions for employees have been made on the public and private level. Efforts by the government to ensure health and safety have been made through standard operation regulations. On the private level, organizations also ensure that their valuable workforce is ready for their operations. Overall, health is an important driver of efficient and happy workplaces, as poor health among employees can be detrimental to performance and the bottom line.

Regarding the workplace, the SanPIN 2.2.2./2.4.1340-03 defines the following conditions: The workplace with a personal computer must be in relation to the window openings apertures so that light fell sideways, more preferably from the left. It is necessary to avoid arrangements of working places in corners of the room. Further, workers should ideally not be facing the wall. The recommended distance from the PC to the wall should be a least 1m. An orientation of the screen and the face towards the window is preferred. Ideally, the PC is positioned in a way that when looking up from the screen, a very distant object can be seen. Not only this can be considered as a pleasant working position, also support for the workers' health is given as it is one of the most effective ways to unload the visual system.

Within the workplace, users should be exposed to microclimate parameters according to the SanPIN 2.2.2./2.4.1340-03. According to this work regulation, the air temperature in the cold period must not be more than 22-24°C. In the warm season 20-25°C are advised. Relative humidity should range between 40-60% and the air velocity should be 0.1m/s. For maintaining optimal values, the heating and air conditioning systems should be used.

It is important to point out that comfort and choice are factors that should be considered for workplace health. Choice is positively correlated with workplace health, so offering a variety of work settings for various work functions (e.g., focus, learning, socializing) also reduces stress and frustration (Augustin, 2009). Choice among workspaces reduces stress caused by noise, crowding, coworkers and discomfort. In regard to the workplace temperatures, one has to remember, that people feel temperatures differently, so allowing control temperature of their own workspace can increase comfort. If a single temperature is selected, lower temperatures, ideally 21.6°C, are associated with reduced sick leave (Witterseh, Wyon, & Clausen, 2004). If individual temperature control is not an option, the building can offer various temperature controlled areas from which to choose. In consideration of atmosphere control, it is recommended to provide filtered air and a clean environment to reduce respiratory problems such as asthma (Kats et al., 2003).

The office space requirements have been defined in the SanPIN 2.2.2./2.4.1340-03 and in the SanPIN 2.09.04.87. Here the hygienic requirements to personal computers and the work organizations as well as the administrative and domestic premises and industrial buildings have been described. The requirements include the are per seat for an adult must be at least 6m^2 and air volume should be at least 20m^3. According to the SanPIN 2.2.1./2.1.1.1278-03, informatics offices need to have a natural and artificial lightning. Rooms with computers should be equipped with heating, air conditioning or forced-air-exhaust ventilations. The floor surfaces should be smooth and non-slippery. The main flow of natural light should be on the left. The noise level should not exceed 50 dBA. Further, any room should have a first-aid kit and a fire extinguisher.

Altering viewing distance, changing the screen setup, ensuring proper lighting and monitoring the ergonomics of the desk environment can help. Furthermore, taking frequent eye breaks while working is very important. It is recommended to practice the "20/20" rule which says to look away from the computer every 20 minutes for 20 seconds to minimize eye-focusing problems and irritation caused by infrequent blinking.

To avoid any electronic malfunctioning, the electronic devices need to be kept up to date, maintained and serviced.

A synthesis of studies, that are linking physical activity to health, was able to show an adverse health effect for low levels of activity. As the software development tasks mainly takes places seated in front a computer, it is advised to compensate the lack of physical activity through possibilities for the employee to exercise in sport facilities provided by the company or to incorporate some sports into an employee's leisure time.

E.2. Environmental Safety

Activities of a person can cause enormous damages to the environment and it is necessary for every individual to keep his or her influence on the environment as low as possible. According to the Federal Law on Environmental Protection No. 7-FZ of 10th January 2002, each citizen has "[...] a right to a favorable environment, everybody shall preserve the nature and the environment, carefully deal with the natural wealth being a basis for the sustainable development, life and activities of the peoples inhabitation the territory of the Russian Federation".

Regarding the software development process of safe messengers, today's computers and laptops and their associated electronic devices only consume comparatively little electricity. The software development process is assumed to mainly take place in office settings without direct exposure to the nature. The environmental safety, hence, lies not primarily in the usage of IT devices. The problem is the production and the disposal of those electronic components and devices, as their fabrication might not always be environmental friendly.

E.2.1. Impact analysis of research object on environment

On the one hand, the technological progress and the safety standards require up to date hardware in the software development progress such as in the development of safe messengers. On the other hand, this need for keeping up renders the life span of hardware down to approximately 2-3 years. Thus, the software developer consumes a considerable amount of raw materials and energy, although while using the hardware, the energy consumption might be rather low.

The initial production phase for producing hardware has a high environmental impact. The use of heavy metals and hazardous chemical incorporated in hardware is the decisive factor in the environmental safety in IT. An analysis by Greenpeace showed that for almost every component in electronic devices, either bromine or plastic PVC are used.

After using the hardware for a couple of years, the devices need to be disposed.

E.2.2. Impact analysis of research process on environment

While developing software, energy is used. The software developer cannot avoid using electricity. However, the worker can make sure that he is using renewable energy sources. Renewable energy is obtained from sunlight, wind, tides, waves or geothermal heat and is naturally replenished on a human timescale.

Each source of renewable energy has unique benefits and costs. As traditional energy produced from coal, oil, and natural gas, also renewable energy sources have some impact on the environment. However, fossil fuels do substantially more harm, than renewable energy sources by most measures, including air and water pollution, damage to public health, wildlife and habitat loss, water use, land use, and global warming emissions.

E.2.3. Protection methods to mitigate the potential damage

With the worldwide sales reaching 1.86 billion devices in 2014 only for the mobile phone market, the ecological footprint of an electronic device may be small, but the cumulative effect is to be quite significant on a global scale. To reduce these impacts, consumers play a central role in sustainable production by purchasing eco-friendly products.

To ensure that the developer him or herself is making the smallest mark possible in the software development progress, hardware that ensures to be sustainably developed could be used. Further, the developer can ensure that while working, he is using electricity obtained from renewable energy sources.

Upon being returned, most electrical and electronic waste can be recycled. Recovered metals such as iron, steel, copper, aluminum and bronze can be melted down to make new metals. Motherboards and elements such as plugs with gold-plated contacts are normally sent to copper foundries that specialize in the recovery of precious and special metals. As for sorted plastic, some can be reused for energy, while some can be recycled. The developer should, if possible, determine before buying a product, that the manufacturer recycles in an exemplary manner. In general, manufacturers need to be held responsible for properly disposing the electric and electronic waste.

E.3. Safety in emergency

Emergencies can create a variety of hazards for workers in the impacted area. Preparing before an emergency plays a vital role in ensuring that employers and workers have the necessary equipment, know where to go, and know how to keep themselves safe when an emergency occurs. Companies are strongly advised to train emergency behavior at least once a year and to inform new employees when they start working.

E.3.1. Identification and analysis of emergency situations, which the research object can create

A workplace emergency is a situation that threatens workers, customers, or the public; disrupts or shuts down operations; or causes physical or environmental damage. Emergencies may be natural or man-made, and may include hurricanes, tornadoes, earthquakes, floods, wildfires, winter weather, chemical spills or releases, disease outbreaks, releases of biological agents, explosions involving nuclear or radiological sources, and many other hazards. Many types of emergencies can be anticipated in the planning process, which can help employers and workers plan for other unpredictable situations. The software development process itself can produce man-made emergencies. In case of wrong implementation, the operational system might shut down, leading customers or even whole organizations unable to operate. Also, electrical malfunctioning can lead to fire outbreaks.

E.3.2. Identification and analysis of emergency situations, which may occur during the research process

The software development process usually takes place in office buildings. As for most people in IT, the greatest danger therefore is from the usual occupational hazards such as slipping, falling and the risk of fire. Books should be easily reachable and not stored on poorly accessible areas. In case of an unforeseen electric malfunctioning such as an electric burn, a fire might break out. Fires can spread out extremely fast. Preventional work based on the minimization of the risk factors is advised. Fire in an office room can lead to material damage and can be life-threatening.

In case of health related incidents such as a heart attack, employees should be able to help one another. The entire staff shall complete a first aid course. A requirement to refresh this course every two years should further be defined.

E.3.3. Protection methods to mitigate the potential damage

Fire prevention is a complex of organizational and technical actions directed towards the safety of employees. The creation of several conditions, though, can reduce the likelihood of a fire to a minimum. Employees awareness to the issue can be raised through special fire emergency training. Manuals and evacuation plans can further help employees in an emergency situation to leave the office building safely until help arrives. To extinguish fires in the initial stage, fire extinguishers shall also be places nearby every office.

To reduce electrical risks to a minimum, all sockets should be protected with differential circuit breakers. To detect emergencies fast and easily, surveillance has proven itself to be very useful. Occupational health nurses, case managers, risk management experts, and physicians could review the workplace and employees to ensure functionality.

Also, employee's health could be monitored. Medical trends, compliance, regulatory requirements, and mitigate absences could be anonymously be tracked and to make informed decisions on how to improve employee health and productivity.

E.4. Workplace design (workplace ergonomics)

An organization that strives to maximize health goes beyond safety requirements (e.g., hazard prevention), offering holistic health services on primary, secondary and tertiary levels, focusing first on primary (i.e., preventative) elements and programs. Primary levels include those methods to avoid occurrence (e.g., removing causes of stress). Secondary levels address symptoms (e.g., monitoring stress), and tertiary levels provide rehabilitation (e.g., time off) (Quick, Murphy, & Hurrell, 1992).

Ergonomic designs can help in preventing musculoskeletal diseases (e.g. back pain, arthritis). Without a proper ergonomic setup, software developers run the risk of back and spine injuries. Problems can include anything from cervical radiculopathy (a compression of the nerve roots in the neck) and bursitis of the shoulder on down to pulled or strained muscles, ligaments and tendons in the lower back. Offering standing desks is a preventative remedy to sitting-induced health problems (Patel et al., 2010).

Beyond these preventative health factors of an ergonomic work place or the right temperature, companies can go a step further by inserting nudging elements in the workplace. For instance, according to Google's Head of People Operations Laszlo Bock (2015), the company tested whether or not swapping out the plates in the cafeteria for smaller ones would result in less food consumption among employees. Even though the food is free and the employees could go back from more, the nudge was effective and resulted in healthier eating habits (Bock, 2015). Other elements can also nudge physical activity, such as providing running trails onsite and the choice of walking desks for those employees who find themselves restless at work (Leblanc, 2016). The consistent element here is choice because, it is important that employees do not feel pressure from the company to make healthy decisions, which could result in additional stress.

E.5. References

- Augustin, S. (2009). Place Advantage.
- Biddle, Fox, K., & Boutcher, S. (2003). Physical activity and psychological well-being: Routledge.
- Bock, L. (2015). Work rules! : insights from inside Google that will transform how you live and lead (First edition. ed.). New York: Twelve.
- Federal Law on Environmental Protection No. 7-FZ of January 10, 2002 Adopted by the State Duma December 20, 2001
- Kats, G., Alevantis, L., Berman, A., Mills, E., & Perlman, J. (2003). The costs and financial benefits of green buildings. A Report to California's Sustainable Building Task Force. USA.
- Palmer, K. T., Harris, E. C., & Coggon, D. (2007). Carpal tunnel syndrome and its relation to occupation: a systematic literature review. Occupational Medicine, 57(1), 57-66.
- Patel, A. V., Bernstein, L., Deka, A., Feigelson, H. S., Campbell, P. T., Gapstur, S. M., ... Thun, M. J. (2010). Leisure time spent sitting in relation to total mortality in a prospective cohort of US adults. American journal of epidemiology, 172(4), 419-429.
- Quick, J. C. E., Murphy, L. R., & Hurrell, J. J. J. (1992). Stress & well-being at work: Assessments and interventions for occupational mental health: American Psychological Association.
- Sharma, M., & Majumdar, P. K. (2009). Occupational lifestyle diseases: An emerging issue. Indian J Occup Environ Med, 13(3), 109-112. doi:10.4103/0019-5278.58912
- Thomsen, J. F., Gerr, F., & Atroshi, I. (2008). Carpal tunnel syndrome and the use of computer mouse and keyboard: a systematic review. BMC musculoskeletal disorders, 9(1), 134.
- Witterseh, T., Wyon, D. P., & Clausen, G. (2004). The effects of moderate heat stress

and open-plan office noise distraction on SBS symptoms and on the performance of office work. Indoor Air, 14(s8), 30-40.

List of Figures

List of Tables

Bibliography

[12] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model (CCMB-2012-09-001). Sept. 2012.

[Ahr14] J. Ahrens. Threema protocol analysis. Mar. 2014. URL: `http://blog.jan-ahrens.eu/files/threema-protocol-analysis.pdf`.

[Bun10] Bundesamt für Sicherheit in der Informationstechnik. The PP/ST Guide. Aug. 2010.

[EFF] EFF. Secure Messaging Scorecard. URL: `https://www.eff.org/node/82654`.

[EMH16] K. Ermoshina, F. Musiani, and H. Halpin. End-to-End Encrypted Messaging Protocols: An Overview. In: Internet Science: Third International Conference, INSCI 2016, Florence, Italy, September 12-14, 2016, Proceedings. Ed. by F. Bagnoli, A. Satsiou, I. Stavrakakis, P. Nesi, G. Pacini, Y. Welp, T. Tiropanis, and D. DiFranzo. Cham: Springer International Publishing, 2016, pp. 244–254. ISBN: 978-3-319-45982-0. DOI: `10.1007/978-3-319-45982-0_22`.

[JO15] J. Jakobsen and C. Orlandi. On the CCA (in)security of MTProto. Cryptology ePrint Archive, Report 2015/1177. `http://eprint.iacr.org/2015/1177`. 2015.

[KBB17] N. Kobeissi, K. Bhargavan, and B. Blanchet. Automated Verification for Secure Messaging Protocols and their Implementations: A Symbolic and Computational Approach. In: IEEE European Symposium on Security and Privacy (EuroS&P). to appear. 2017.

[Par16] N. I. A. Partnership. Protection Profile for Application Software Version 1.2. Apr. 2016. URL: `https://www.niap-ccevs.org/Profile/Info.cfm?id=394`.

[Rot+15] C. Rottermanner, P. Kieseberg, M. Huber, M. Schmiedecker, and S. Schrittwieser. Privacy and Data Protection in Smartphone Messengers. In: Proceedings of the 17th International Conference on Information Integration and Web-based Applications & Services. iiWAS '15. Brussels, Belgium: ACM, 2015, 83:1–83:10. ISBN: 978-1-4503-3491-4. DOI: `10.1145/2837185.2837202`.

[Ung+15] N. Unger, S. Dechand, J. Bonneau, S. Fahl, H. Perl, I. Goldberg, and M. Smith. SoK: Secure Messaging. In: 2015 IEEE Symposium on Security and Privacy. May 2015, pp. 232–249. DOI: `10.1109/SP.2015.22`.