Пловдивски университет „Паисий Хилендарски”
Факултет по математика и информатика
Катедра „Компютърна информатика”

Тони Пламенов Каравасилев

СОФТУЕРНА РАМКА ЗА КРИПТОГРАФСКИ УСЛУГИ

А В Т О Р Е Ф Е Р А Т
на дисертационен труд за присъждане на
образователна и научна степен „доктор“

Област на висше образование:
4. Природни науки, математика и информатика
Професионално направление: **4.6. Информатика и компютърни науки**
Докторска програма **Информатика**

Научни ръководители:
проф. д-р Елена Петрова Сомова
доц. д-р Светослав Христосов Енков

Пловдив, 2021 г.

Пловдивски университет „Паисий Хилендарски”
Факултет по математика и информатика
Катедра „Компютърна информатика”

Тони Пламенов Каравасилев

СОФТУЕРНА РАМКА ЗА КРИПТОГРАФСКИ УСЛУГИ

А В Т О Р Е Ф Е Р А Т
на дисертационен труд за присъждане на
образователна и научна степен „доктор“

Област на висше образование:
4. Природни науки, математика и информатика
Професионално направление: **4.6. Информатика и компютърни науки**
Докторска програма **Информатика**

Научни ръководители:
проф. д-р Елена Петрова Сомова
доц. д-р Светослав Христосов Енков

Пловдив, 2021 г.

Дисертационният труд е обсъден и насочен за защита от катедрения съвет на катедра „Компютърна информатика“ на Факултета по математика и информатика при Пловдивския университет „Паисий Хилендарски“ на 23.09.2021 г.

Дисертационният труд „Софтуерна рамка за криптографски услуги“ съдържа 165 страници, от които 9 приложения от 9 страници.

Библиографията включва 117 заглавия.

Списъкът на авторските публикации се състои от 5 заглавия.

Материалите по защитата са на разположение на интересуващите се в библиотеката на Факултета по математика и информатика на ПУ „Паисий Хилендарски“, бул. „България“ № 236, Пловдив.

Защитата на дисертационния труд ще се състои на 20.12.2021 г. от 12:00 ч. онлайн (ще бъде обявено в сайта на ПУ/ФМИ).

Автор: **Тони Пламенов Каравасилев**

Заглавие: **СОФТУЕРНА РАМКА ЗА КРИПТОГРАФСКИ УСЛУГИ**

Тираж 50 бр.

Пловдив, 2021 г.

Съдържание

СЪКРАЩЕНИЯ

ASCII	–	American Standard Code for Information Interchange
CC0	–	Creative Commons Zero v1.0 Universal
CERN	–	European Organization for Nuclear Research
CLI	–	Command-Line Interface
DOI	–	Digital Object Identifier
ISBN	–	International Standard Book Number
ISSN	–	International Standard Serial Number
IV	–	Initialization Vector
MAC	–	Message Authentication Codes
MDC	–	Message Detection Codes
MIT	–	Massachusetts Institute of Technology
PHP	–	Hypertext Preprocessor
PHS	–	Password Hashing Scheme
SJR	–	Scientific Journal Rankings
SK	–	Secret Key
ПУ	–	Пловдивски университет „Паисий Хилендарски“
ФМИ	–	Факултет по математика и информатика

УВОД

Актуалност на проблема

В днешно време е трудно да си представим софтуерна система без автентикация, генериране на идентификатори, криптиран трафик, механизми за размяна на сигурни съобщения, проверки за целостта на информацията или методи за безопасно складиране на данни от тип парола, кредитна карта или дори файлове, като изображения или документи. В следствие на тази нужда от различни криптографски услуги и примитиви, изниква проблемът с имплементирането на сложни взаимодействия или комбинации от различните типове алгоритми за информационна сигурност. Това може да забави разработката на дадена софтуерна платформа или да усложни поддръжката ѝ.

Цели и задачи на дисертационния труд

Основната цел на дисертационния труд е да се разработи обектно-ориентирана софтуерна рамка за криптографски услуги за употреба в процесите на проектиране, реализиране, поддържане и тестване на сигурни софтуерни системи от проектанти и програмисти. За реализирането на такава софтуерна рамка трябва първо да се проектира цялостен криптографски модел и да се предложи подходяща методика за защита на информацията, а след това на тяхна база да се реализират унифицирани колекции от конкретни криптографски услуги, протоколи и примитиви.

За да се постигне основната цел, е нужно да бъдат решени следните ***задачи***:

Задача 1. Проучване и анализ на теоретичните основи на криптографията и криптологията, включващо обзор на съвременните международни стандарти за криптографски системи, качествени подходи за реализация на сигурна мрежова комуникация и добри практики за защита на информацията при пренос, съхранение и обработка;

Задача 2. Проектиране на цялостен криптографски модел на базата на криптографски услуги, протоколи, примитиви и йерархии от обекти, както и на методика за интеграция на криптографски услуги в реална среда;

Задача 3. Изграждане на обектно-ориентирана софтуерна рамка за криптографски услуги на базата на дефинирания криптографски модел и софтуерни абстракции;

Задача 4. Тестване на изградената специализирана софтуерна рамка за съвместимост, производителност и качество, както и анализ на резултатите, включително спрямо съществуващите конкурентни решения или процедурни библиотеки.

Етапи на дисертационното изследване

Постигането на основната цел предполага провеждането на следните ***етапи***:

1. Запознаване с основните криптографски функционалности, нужни за изграждането на информационни системи и особеностите при реализацията на криптографски услуги или сигурни мрежови протоколи;
2. Извеждане на добри методологии, практики и съвети за коректното интегриране на мерките за мрежова и информационна сигурност в помощ на проектанти и програмисти на уеб сайтове, приложения, системни конзоли и мрежови архитектури;
3. Проектиране, реализация, тестване и техническо документиране на създадената софтуерна рамка за криптографски услуги;
4. Анализ на приложимостта на изградения абстрактен криптографски модел и посочване на бъдещи насоки за усъвършенстването му.

Контекст на изследването

Основната сфера на приложение, която ще бъде проследена в настоящия научен труд е изграждането и интегрирането на криптографски услуги с цел разрешаване на проблеми в сферата на информационните технологии и информационната сигурност.

Разработената специализирана софтуерна рамка за криптографски услуги е публикувана онлайн под името CryptoMañana (или CryptoManana, дефинирано като ASCII кодиран синоним на разработката) като отворен код под софтуерния лиценз на MIT (Massachusetts Institute of Technology) и е изцяло разработена чрез езика за програмиране PHP.

Структура на дисертационния труд

Представеният дисертационен труд се състои от увод, четири глави и заключение, списък на използваната литература, списък на авторс-

ките публикации по темата, приложения и декларация за оригиналност. Съдържа разработена методика за реализация и интеграция на криптографски услуги чрез цялостна софтуерна рамка за осигуряване на сигурността на съвременните информационни системи, както и анализ на качеството на използвания криптографски модел на базата на съществуващи разработки реализирани на различни езици за програмиране и видове платформи.

В **Първа глава** се разглеждат теоретичните основи на криптографията и различните видове криптографски системи, както и използваните в съвременната практика стандарти. Направен е обзор на най-ефективните подходи за проектиране, реализиране и документиране за софтуерни рамки.

Във **Втора глава** се разглеждат в детайли процесите по проектиране на абстрактния криптографски модел и специализираната софтуерна рамка за криптографски услуги. Тази глава включва функционален анализ и сравнение на приложимостта на проектираната софтуерната рамка спрямо група от конкурентни софтуерни разработки за различни платформи.

В **Трета глава** се разглеждат в детайли реализацията, тестването, документирането и публикуването на софтуерната рамка за криптографски услуги. Тя включва всички технически детайли по изграждането на нейния цялостен криптографски модел под езика за програмиране PHP.

В **Четвърта глава** се дефинират абстрактни методологии за подсигуряване на информационни системи, изцяло реализуеми чрез интеграцията на изградената софтуерна рамка за криптографски услуги. Тази глава включва обзор на най-ефективните практики и съвети за постигане на цялостна защита на данните по време на техния пренос и съхранение, както и разглежда редица алтернативни методи за постигане на сигурна комуникация между множество страни.

В **Заключението** са систематизирани получените резултати и са изброени научно-приложните и приложните приноси на дисертационния труд. Очертани са бъдещите насоки за развитието на софтуерната рамка и нейния абстрактен криптографски модел.

Основният текст на изследването (общо 165 страници) е придружен от **9 приложения**, поместени на 9 страници, разработени или подобрени в процеса на работата по дисертационния труд. Списъкът на **използваната литература** съдържа 117 заглавия, от които 0 на кирилица, 117 на латиница, включително и 29 уеб-базирани източници.

Списъкът на авторските публикации се състои от 5 излезли от печат заглавия. От тях всички 5 са публикации на английски език.

ГЛАВА 1. ОБЗОР НА КРИПТОГРАФСКАТА ОБЛАСТ

1.1 Въведение

Криптографията е една от най-древните науки, нейната история наброява над няколко хиляди години. В днешно време криптографията се счита за клон същевременно и на математиката и информатиката, като е тясно свързана и с теорията на информацията, компютърната сигурност, мрежовите комуникации и инженерното дело.

1.2 Основни понятия и терминология

Криптографията се използва още от древността и намира огромно приложение в областта на компютърните науки в днешно време. Мрежовата сигурност, компресирането, интернет достъпа, заключването на персоналните компютри, съхраняването на лични данни, извършването на онлайн плащания и криптирането на секретни съобщения между множество страни са немислими без тази наука. Криптографията, заедно с криптоанализа на използваните алгоритмите в една система, се включват в науката криптология. Криптологията представлява наука, обхващаща едновременно създаването, анализирането, подобряването и разбиването на криптографски съобщения [Vaudenay`2005].

Представянето на данните в компютърен формат се нарича кодиране на символни данни (character encoding) [Nelson`1995]. Кодирането на данни се използва за представянето на множество от символи чрез някаква кодираща система, но без цел загуба на четимост или прикриване на информацията. Представянето на дадени данни в друго кодиращо множество се нарича кодиране (encoding). Връщането на дадено множество към първоначалното му представяне (множество) се нарича декодиране (decoding). Важно е да се отбележи, че това преобразование на данни не е с цел прикриване на информацията, а с цел представянето ѝ по разбираем начин в даден контекст или оптималното ѝ пренасяне в мрежа.

1.3 Генератори на данни и източници на случайност

Практическата нужда от генератори на псевдо-случайни данни и приложимостта им в криптологията е огромна. Общата езикова дефиниция за случайност представлява липсата на шаблон или каквато и да е предсказуемост в дадено събитие или количество информация [Goldreich`1998]. В допълнение на тази характеристика, случайната последователност от събития, символи или стъпки също трябва да бъде из-

цяло непредсказуема или без никакъв видим ред от комбинация или явни зависимости. В практиката, при тестване на огромно количество от непредсказуеми по себе си събития, известни още като опити или проби, все още е възможно да се появят зависимости или да се идентифицира неочакван модел на взаимодействие между тях.

В математиката, статистиката и информатиката, за случайна променлива се счита присвояването на числова стойност към всеки възможен изход или резултат за дадено събитие, включен в обхвата или областта на това събитие [Smart`2013]. Като термин в информатика, случайността се счита за информационен шум (white noise или static sound), който представлява данни без някакво значение или дори безсмислено количество от информация. Интегрирането на източници на случайни данни в софтуерните системи се е превърнало в необходимост и е дълбоко включено, както в цикъла на разработка, така и за ежедневното функциониране на всяка съвременна софтуерна платформа. Реализирането на практически източник на случайност в изчислителните науки и софтуерните технологии се базират изцяло на алгоритмите от теорията на числата и съществуващите методи за изграждане на достатъчно произволно изглеждащи числови последователности.

В софтуерните технологии, употребата на физически или изчислителни източници на ентропия става чрез така наречените генератори на случайни числа или двоични данни. Независимо от избрания източник на случайност за дадената система или платформа, реализацията на алгоритми за генерация на числа е в основата на възпроизвеждане на посложни форми на информация като имена, телефонни номера или дори изображения. Генерирането на случайни числа намира приложение в хазарта, компютърните симулации, криптографията и други места, където е нужен непредвидим краен резултат.

В практиката, те най-често се използват за генериране на висококачествени идентификатори, временни пароли за достъп от устройства (tokens), потребителски пароли, ключове за криптографски системи, криптографска сол и временни автентикиращи криптографски данни (cryptographic nonce или authentication word). Най-добрите практики за киберсигурност изцяло разчитат на качеството на генерираните данни и на липсата на повторяемост или преизползването им в дадената система. Поради това е възможни при бъдещи проблеми с източника на тези данни да се компрометира цялостната сигурност на една платформа и използваните криптографски алгоритми в нея.

Голяма част от съвременните централни процесори добавят и стандартизирани източници на ентропия в архитектурата си, достъпни

директно от ядрото на операционната чрез колекция от специфични инструкции. Те използват комбиниран подход, като записват генерирания информационен шум в така наречения временен буфер за ентропия (entropy pool) и подават всяка една стойност към отделните процесорни нишки като инициализираща такава за софтуерни криптографски алгоритми за генериране на псевдо-случайни числа в даден системен процес.

1.4 Еднопосочна криптография

Свойствата на еднопосочната криптографска обработка на данните в съвременните системни намират огромна приложимост в разработката на хардуерни и софтуерни решения. Хеш функцията (hash function) представлява еднопосочна математическа функция, която работи с входящи данни (съобщение) от произволна дължина [Katz`2007]. Изходните данни на такава функция връщат като резултат от изпълнението ѝ информация с фиксирана дължина, най-често представяна като низ или число. Действието по преобразуване на входящите данни се нарича хеширане или раздробяване на информацията.

При този тип функции се получава компактно представяне на входната последователност, което наричаме още цифрова сигнатура, отпечатък или извлечение (message digest или digital fingerprint). Изходния низ представлява число, което се нарича още хеш код, хеш резултат, хеш стойност или просто хеш. При тези функции дори промяната на един символ е от значение, при което изходния продукт се променя драстично и се цели гарантирането на уникалност на изходящия хеш спрямо дадена област от входящи данни.

Хеш функциите се използват за удостоверяване на източника (MAC), за удостоверяване на целостта на данните (MDC), за сигурното сравнение на данни или за генериране на уникални деривативни стойности спрямо определен ресурс [Menezes`1996]. При използването на еднопосочна криптография за удостоверяване на целостта на данни се генерира хеш за съобщение, което ще бъде изпратено през мрежова връзка или съхранено на дадено физическо устройство. От друга страна, при употреба на хеш функции за удостоверяване на източника се включва и друг тип криптографски алгоритми от асиметричен характер. При използването за хеш функции за проверка на целостта на данните, се генерира хеш на съобщението за изпращане, но тук в допълнение този хеш се криптира асиметрично преди преноса му с цел удостоверяване на изпращача му.

В съвременните информационни системи съществува и допълнителен тип от еднопосочни криптографски преобразования, реализиран с цел сигурно съхранение на чувствителна или тайна информация, а също

и с възможност за сигурно генериране на множество от ключове деривати спрямо задени входни данни. Този тип алгоритми се наричат дериватни или деривативни, като са реализирани чрез многократни итерации и приемат множество от начални ключове, както и дори допълнителни параметри за ограничаване на използваните ресурси на дадената система. Алгоритмите, реализирани за генериране на множество от ключове, са проектирани да работят бързо и всеки път да създават едни и същи стойности спрямо подадените входящи параметри или ключове.

Съществува и по-специфична вариация на тези дериватни алгоритми с възможност за подаване на различни параметри за ограничаване на ресурсите, използвани за извършване на изчисления, още наричани хардуерно-устойчиви дериватни хеш функции или схема за хеширане на пароли (Password Hashing Scheme) [Aumasson`2017]. Те намират най-голяма употреба в практиката, както за сигурното съхранение на пароли, така и за потвърждаване на идентификацията на потребители, системи или платформи. Използването на дериватни функции, устойчиви на хардуерни атаки и коректното конфигуриране на параметрите им за лимитиране на използваните ресурси при обработка на данни, прави почти невъзможно генерирането на дъгови таблици или изпълнението на атаки от тип „груба сила“ [Subangan`2019].

1.5 Симетрична криптография

Симетричната криптография включва криптографските алгоритми, базирани на употребата на един и същ секретен ключ за криптиране на прав текст и за декриптиране на шифрограма. Този тип криптосистеми могат и да приемат различен на брой допълнителни параметри за усложняване на процеса на обработка. За да функционира коректно една криптографска система от този тип, се изисква всяка една от двете страни да притежава дадения секретен ключ (secret key) или така наречена взаимна тайна [Katz`2007]. Този ключ трябва да остане недостъпен за трети страни, в противен случай ще бъде изцяло нарушена секретността на данните.

За разлика от еднопосочната криптография, този тип алгоритми целят осигуряването на възможност за възвращаемост на първоначалното съобщение за бъдещо ползване, но единствено от заинтересованите странни в дадената комуникация. Поради този факт, често се използват за скриване на лични данни, които е нужно да бъдат използвани за извършване на плащания, проверка на документи, лични изображения или възстановяване на достъп до даден потребителски профил.

Криптосистемите от този тип могат да бъдат реализирани по коренно различен начин една от друга, но се характеризират със симетричността на операциите при трансформация на данни. Основните два вида

симетрични алгоритми са потоковите шифри (stream ciphers) и блоковите шифри (block ciphers) [Menezes`1996]. Потоковите шифри криптират отделните знаци на едно текстово съобщение изцяло последователно, като използват трансформации на база знак по знак, но и често включват последователна обработка на база последно обработения знак. Блоковите шифри обработват данните, като ги разделят на група или блок от знаци с фиксирана дължина и прилагат криптографски трансформации над всеки един техен фрагмент от цялостната информация в дадена посока.

Съвременните криптографски симетрични системи. базирани на блокова обработка, притежават няколко допълнителни унифицирани параметри за конфигурация, отделно от ключа, използван за криптиране. Най-важният параметър след ключа за криптиране е режима на работа (mode of operation) или още наричан алгоритъм за обработка на блокове. Той задава логиката на операцията прилагана над един блок от данни и свързаността или зависимостта между всеки един отделен фрагмент спрямо останалите. За първият блок е нужен допълнителен параметър към използваната криптографска система, наречен инициализиращия вектор (initialization vector). Важно е да се отбележи, че след като блоковите шифри работят чрез разделяне на данните на групи с фиксирана дължина, то последният блок може да се получи с по-малка дължина. В този случай се прилага така нареченото подравняване (padding), което интегрира определен стандарт за допълване на информацията.

1.6 Асиметрична криптография

Реализацията на сигурна мрежова комуникация по публичен канал е прекалено трудна без наличието на асиметричните криптографски алгоритми. Асиметричната криптография включва криптографските алгоритми базирани на употребата на различни ключове за операциите криптиране и декриптиране. Тези системи се наричат още асиметрични криптосистеми или системи с публичен ключ.

За да функционира коректно една криптографска система от този тип, се изисква страната изпращач да притежава даден публичен ключ (public key), а страната получател да притежава даден частен ключ (private key) [Aumasson`2017]. Публичният ключ се използва за криптиране и може да се разпространява свободно дори за повече от един изпращач. Частният ключ от своя страна, трябва да се пази в тайна и да бъде притежаван само от страна на получателя. Важно е да се отбележи, че комбинацията от публичен и частен ключ е предварително избрана и генерирана като криптографско свързана двойка. Математически е възможно генерирането на множество ключове спрямо еднакъв частен или

публичен ключ, но практически това е лоша практика и изцяло разрушава сигурността на установените стандарти за асиметрично криптиране.

Тези системи обработват данните като общ единичен блок с ограничен максимален размер, който винаги е по-малък от дължината на поддържания ключ. Както при симетричните криптографски системи, тук също се налага употребата на стандарти за допълване на данните (padding) до дадената дължина, изисквана от алгоритъма. Въпреки, че е възможно ръчното обработване на по-големи данни чрез деленето им на достатъчно малки блокове, криптирането им чрез асиметрични алгоритми и слепването на получените шифрограми като краен резултат, това не се препоръчва и е считано за лоша практика. Интегрирането на такъв тип логика в една система създава възможности за експлоатация на различни проблеми, породени от обработката, намалява общата производителност и изцяло противоречи на целта, за която асиметричните системи са изначално проектирани.

Асиметричните криптографски системи се използват в практиката не само за криптиране на важна информация, но и за реализацията на алгоритми за размяна на ключове между две страни, имплементации за цифрово подписване на документи и като лесен начин за премахване на риска от подслушване по време на мрежово комуникиране. Главната цел на асиметричните криптосистеми е да криптират малко количество от данни за изпращане по публичен канал [Schneire`2013]. Тези системи обработват данните като общ единичен блок с ограничен максимален размер, който винаги е по-малък от дължината на поддържания ключ. Например, такива стандарти често се използват за реализирането на сигурно предаване на таен ключ между две страни, използван за инициализиране на симетрична криптографска система за подсигуряването на бъдещата комуникация.

ГЛАВА 2. КРИПТОГРАФСКИ ОБЕКТНО-ОРИЕНТИРАН МОДЕЛ

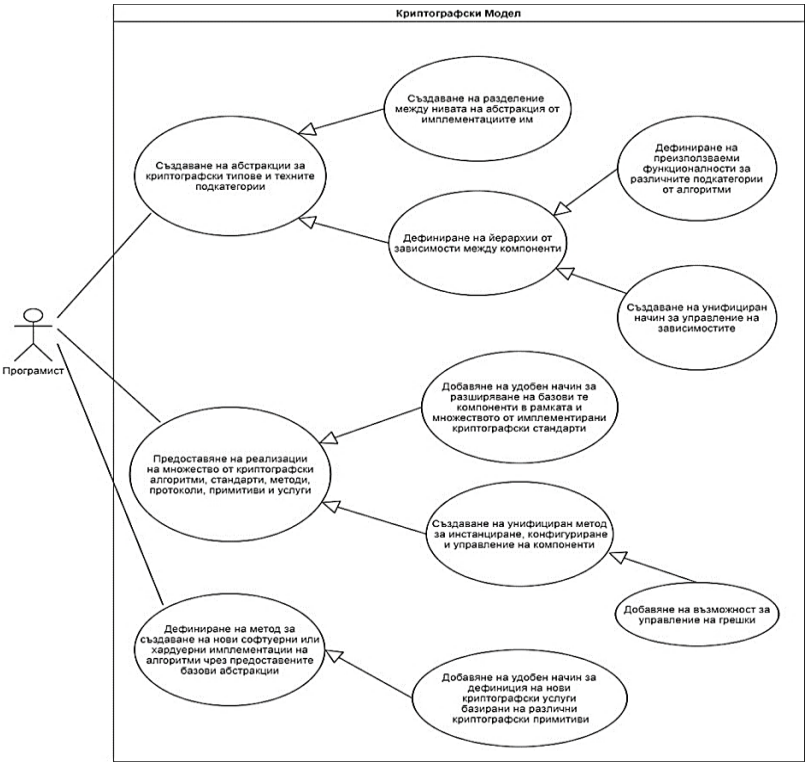
2.1 Проектиране на криптографски модел

Когато се налага имплементирането и употребата на множество от различни криптографски алгоритми в дадена софтуерна система, изскача нуждата от унифицирана базова спецификация и разделение на функционалността на криптографските системи по типове, което изцяло липсва в повечето криптографски библиотеки или рамки.

Като първа стъпка за разрешаването на тези проблеми, е нужно дефинирането на модел, съдържащ разделение между различните типове

алгоритмите и създаване на базова спецификация на функционалността. Изискванията (Фигура 1) за такъв тип представяне трябва да включват:

- Дефиниране на обектно-ориентирани абстрактни класове за характеризиране на различните типове алгоритми спрямо нивото им на сложност и броя на техните зависимости спрямо други компоненти;
- Планиране на удобен метод за създаване на нови имплементации на алгоритми чрез разширяването на базовите обекти;
- Вграждане на унифициран набор от параметри и функционалности за различните компоненти, с лесна възможност за разширяване или променяне избрания обект;
- Дефиниране на обектни йерархия за различните подкатегории от даден тип криптографски примитиви, протоколи, методи или услуги;
- Вграждане на разделение между абстракцията и имплементацията на компоненти.

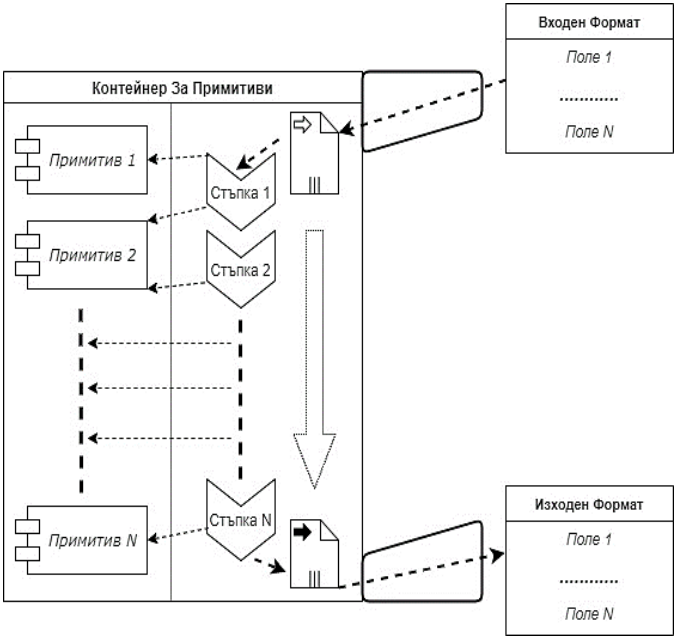


Фигура 1. Основни изисквания за функционалността на криптографския модел

Проектирането на изцяло обектно-ориентиран криптографски модел за лесно интегриране и създаване на криптографски услуги, включва нуждата от въвеждане на следното разделение на компонентите при дефинирането им като категории в модела:

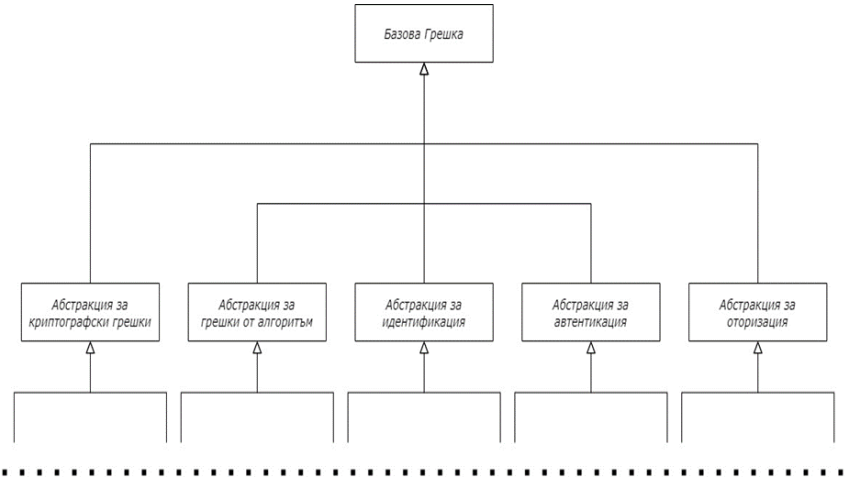
- Източник на псевдо-случайност;
- Хеш функции;
- Симетрични криптографски системи;
- Асиметрични криптографски системи;
- Криптографски протоколи;
- Криптографски структури от данни;
- Услуги за сигурно генериране/изтриване на информация.

Поради разнообразието от съществуващи или възможни бъдещи криптографски услуги и формати на обработка, в криптографският модел трябва да се проектира гъвкава базова абстракция под формата на контейнер за примитиви с минималистична спецификация (Фигура 2). Зависимостите за всеки отделен протокол, метод или услуга трябва да бъдат специфицирани чрез отделни интерфейси, които да могат да бъдат преизползвани за подобни по характер компоненти. По този начин се избягва създаването на огромна и трудна йерархия от протоколи, и се създава унифицирана спецификация за имплементация на услуги в рамката.



Фигура 2. Абстрактно представяне на криптографски услуги

В контекста на криптографския модел се налага допълнителното създаване на абстракция за криптографски грешки чрез която да се дефинира йерархия от конкретни семантични грешки (Фигура 3) за специализирани цели от дадената област. Важно е да се отбележи, че тези грешки ще се използват за прихващане на семантични или технически проблеми в реализацията на конкретните компоненти на криптографския модел. Освен това, те трябва да са налични за програмиста с цел директна употреба по време на бъдещи реализации на криптографски стандарти под рамката. В допълнение, препоръчително е да се добави и възможност за маркиране на грешки при процеси от високо ниво, като например, такива при реализацията на идентификация, автентикация или оторизация.



Фигура 3. Абстрактно йерархично разделение на криптографски грешки

2.2 Проектиране на софтуерна рамка

Софтуерните библиотеки сами по себе си предлагат удобен начин за преизползване на даден тип функционалност, но по никакъв начин не дефинират или предоставят абстрактна структура на компонентите, включени в тях [Vogel`2011]. Благодарение на съществуването на софтуерните рамки, е възможно дефинирането на унифицирана структура за всички модули или библиотеки, независимо от техния тип или предназначение. Това спестява време и усилия, но също премахва и тежката работа по началното създаване на дадена софтуерна система с различни по вид функционалности или характеристики.

В контекста на проектираната разработка, крайният потребител се нуждае от специализирано решение за работа с криптографски функционалности и стандарти. Това определя типа на проектираното решение

като специализирана софтуерна рамка за криптографски услуги. Характеризирайки този софтуерен фреймуърк като такъв за работа криптографския контекст, се появява нуждата от практически подбор на неговите възможности и покритие.

За да се избере конкретен набор от функционалности за имплементация над проектирания абстрактен криптографски модел, трябва да се направи предварителен анализ на най-често използваните криптографски услуги и стандарти в практиката. Най-честите изисквания за защитаване на една система обхващат наличието на компоненти за:

- Сигурно генериране на идентификатори;
- Сигурно генериране на временни пароли за достъп;
- Съхранение на пароли за автентикация в криптиран формат;
- Съхранение на чувствителна информация в криптиран формат;
- Сигурно сравнение на информация за авторизация;
- Проверка за целостта на информацията;
- Проверка на източника на информация;
- Криптиране на информация за сесия;
- Криптиране на комуникация между устройства;
- Управление на множество от криптографски ключове;
- Размяна и генериране на ключове;
- Цифрово подписване на информация;
- Сигурно заличаване на информация.

В контекста на криптографските услуги, е изключително важно да вземем предвид технологичните пречки за успешната реализация на дадената функционалност под дадена платформа. Повечето проблеми от такъв характер могат да бъдат решени чрез софтуерно имплементиране на липсващи стандарти или конструкции на езика. Друга важна част от планирането включва избора на видовете тестове за реализация, което може да е напълно различно спрямо типа на разработката. В контекста на софтуерната рамка за криптографски услуги, най-подходящо би било имплементирането на компонентни тестове за детайлна проверка на имплементираните функционалности [Crispin`2008].

Поради универсалността на проектираното от нас абстрактно решение за дефиниране на криптографски модел и избраните функционални изисквания за софтуерната рамка, може да се разшири обхвата за сравнение на приложимостта на подбраните функционалности. Избраните софтуерните рамки, библиотеки, архитектури и пакети спрямо които ще се извършва сравнението на нашата спецификация, ще включват и такива за различни платформи или езици за програмиране. Резултатите от приложното сравнение изцяло потвърждават стойността на планираните функционалности и концептуални решения в разработката.

Това еднозначно утвърждава качеството на планираната архитектура за софтуерната рамка за криптографски услуги и проектирания ѝ обектно-ориентиран криптографски модел. Отличителните черти на проектираната рамка спрямо останалите разработки включват:

- Въвеждането на обектно-ориентиран стил на компонентите, който не е наличен за криптографските възможности в самия език за реализация;
- Наличието на гъвкав, унифициран и абстрактен криптографски модел;
- Дефинираното йерархично разделение за криптографските примитиви;
- Планираната реализация на достатъчен брой примитиви и услуги;
- Създадените изисквания за голяма съвместимост, тестваемост и поддръжка.

При успешното имплементиране на този набор от алгоритми и стандарти в една криптографска софтуерна рамка, наличната функционалност би била достатъчна за практическата ѝ употреба в реална среда. В допълнение, е препоръчително да се създаде и удобен начин за инициализиране, управление и конфигуриране на различните компоненти в разработката.

Важно е да се вземе предвид и нуждата от дефинирането на други вътрешни типове компоненти в реализацията на криптографския модел. Наличието на такива обекти е от изключителна важност за платформената съвместимост при обработката на информацията спрямо поддържаните кодиращи таблици в системата или достъпните криптографски имплементации алгоритми на ниво операционна система.

2.3 Подбор на технологични способи за разработка

Последната стъпка от фазата на проектирането включва практическия подбор на нужните технологични способи за реализацията на софтуерната рамка за криптографски услуги. В тази част на дисертационния труд в направен обзор на най-важните технически процеси, концепции и решения, необходими за реализацията на разработка от тип софтуерна рамка. Разгледаните техники и стандарти включват:

- Интеграция на обектно-ориентирано моделиране;
- Прилагане на принципите за обектно-ориентираното програмиране;
- Решаване на практически проблеми чрез софтуерни шаблони за дизайн;
- Прилагане на инструменти за проектиране и документиране;
- Преглед на добрите практики в моделирането на софтуер.

ГЛАВА 3. СОФТУЕРНА РАМКА ЗА КРИПТОГРАФСКИ УСЛУГИ

3.1 Реализация на софтуерната рамка

Специализираната софтуерна рамка за криптографски услуги CryptoMañana (Фигура 4) е реализирана на базата на абстрактен скелет, който притежава имплементация на множество от компоненти за съвместимост в ядрото си. Рамката е имплементирана изцяло в обектно-ориентиран стил. Тя използва собствено изолирано наименование в пространството от имена (namespace) на PHP, за да се избегнат всякакви конфликти с имената на базовите конструкции на езика или спрямо паралелната инсталация на други пакети в даден проект.

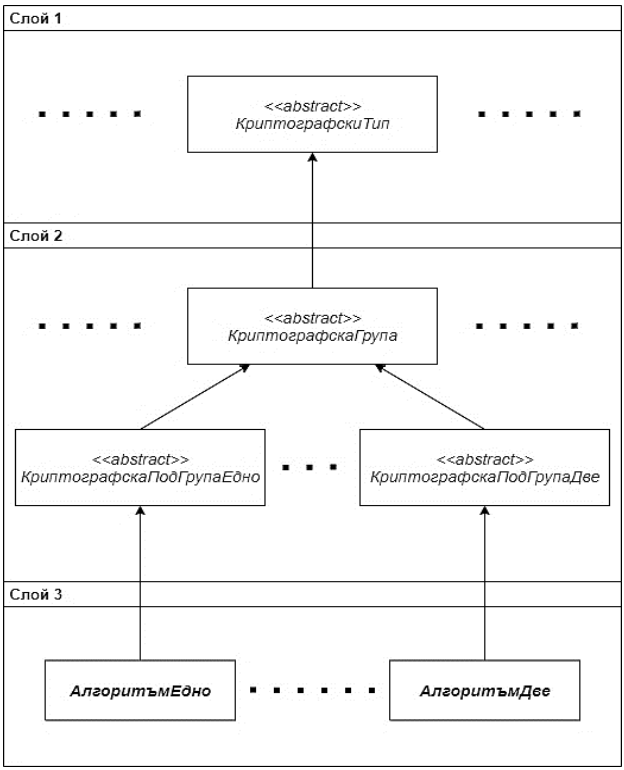
Фигура 4. Лого на реализираната софтуерна рамка за криптографски услуги

Ядрото на самата рамката предоставя огромно количество от абстракции, спецификации и преизползваеми вътрешни компоненти. Имплементираният скелет на рамката се базира на вътрешно абстрактно йерархичното разделение, интерфейси за унифицираност при дефинирането на компоненти и множество от реализации на шаблони за дизайн, като например сек или абстрактна фабрика.

Техническите изисквания за интеграцията на софтуерна рамка за криптографски услуги включват поддръжката на PHP версиите 5.5, 5.6, 7.0, 7.1, 7.2, 7.3, 7.4 и 8.0. Това подсигурява коректната ѝ работа под всички версии на езика между 2013 г. и 2021 г, което е подкрепено от реализираните компонентни тестове в рамката. Важно е да се отбележи, че тази разработка е реализирана с цел коректно функциониране под Windows, Linux и macOS базирани системи, независимо дали те са от 32-битова или 64-битова версия.

Имплементираните компоненти в разработката изрично избягват употребата на външни библиотеки с цел намаляване на зависимостите, но ако определени от тях са налични за текущата система, софтуерната рамка ще ги използва за обработка с цел ускорение на производителността [Wiegers`2013].

Реализацията на криптографският модел в софтуерна рамка за криптографски услуги CryptoMañana се разделя на три семантични части (Фигура 5), разделени на абстрактен тип, опционален подтип и конкретна имплементация на алгоритъм или услуга.



Фигура 5. Структура на криптографски обектно-ориентиран модел

Първото ниво обхваща представянето на отделните главни криптографски типове алгоритми в рамката на ниво абстрактно разделение и дефинира определен базов подход за бъдещо добавяне на нови основни категории. Второто ниво на разделение обхваща абстрактните подтиповете и дефинира различните възможности или характеристики за определените главни криптографски типове. Важно е да се отбележи, че това ниво на абстрактност може да се разделя на многобройни поднива спрямо сложността на представеното криптографско семейство от алгоритми, но не трябва да предлага начин за инициализация на компонентите от този слой. Третото ниво представлява колекция от конкретни реализации на криптографски примитиви и услуги, изцяло разширявайки алгоритми от даден подтип, които са налични във второто ниво на модела.

Семантичното разделение в такъв тип йерархично представяне позволява лесното имплементиране на бъдещи услуги и примитиви в софтуерната рамка, както и задължава програмиста да спазва спецификацията на криптографския модел. Реализацията на тези три нива позволява конкретното специализиране на криптографския модел. В обобщен вид, модулното разделение на полетата от имена в софтуерната рамка за криптографски услуги обхваща:

- Ядро;
- Модул за съвместимост;
- Колекция от грешки;
- Фабрики за инициализация;
- Генератори на данни;
- Хеш функции;
- Симетрични шифри;
- Асиметрични шифри;
- Криптографски услуги базирана на източник на псевдо-случайност;
- Криптографски услуга базирана на примитиви;
- Структури от данни.

В допълнение към тези модули е дефинирана отделна колекция от тестове, копираща структурата на рамката за по-лесната им разработка. Избраната архитектура и разделение на компонентите е дефинирана с цел по-удобната и интуитивна работа на крайния потребител със обектите от софтуерната рамка. Извършена работа по разработването на софтуерната рамка е под средата JetBrains PhpStorm и платформата за виртуализация Docker.

3.2 Тестване на софтуерната рамка

След успешната разработка на софтуерната рамка за криптографски услуги CryptoMañana, тя беше качена на платформата GitHub като публичен проект с отворен код под лиценза MIT (https://github.com/TonyKaravasilev/CryptoManana). По този начин, софтуерната рамка за криптографски услуги е изцяло достъпна онлайн и позволява безплатна ѝ употреба в лична или корпоративна среда.

Софтуерната разработка притежава включено кратко упътване за инсталация, файл с история на промените, лицензен файл, CLI (Command-Line Interface) програма за проверка на системата и включен комплект от всички разработени тестове за рамката. В допълнение, тя е публикувана и в свободната платформа Zenodo, която е разработена от Европейската програма OpenAIRE и е управлявана от CERN за споделяне на научни разработки. За по-лесно цитиране е създаден собствен DOI

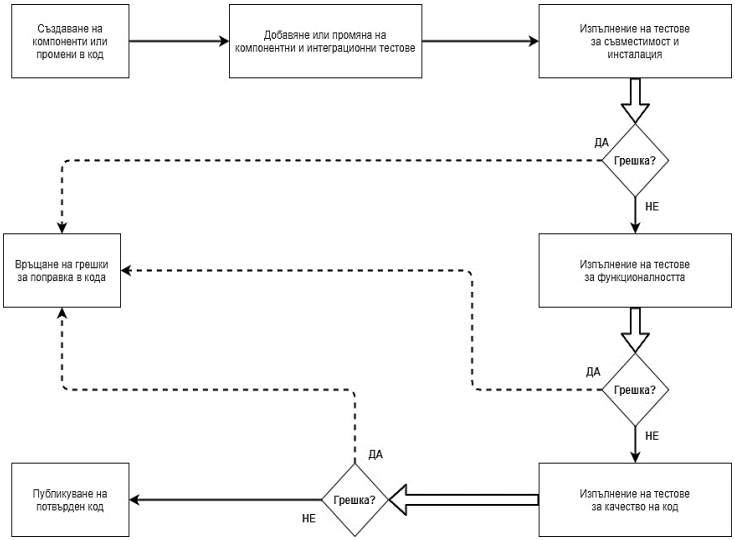
регистриран номер за проекта чрез директна интеграция с Zenodo (https://doi.org/10.5281/zenodo.2604693).

В допълнение, за софтуерната разработка е създаден официален сайт (https://cryptomanana.karavasilev.info) чрез платформата GitHub Pages и генератора на статични сайтове от описателен код Jekyll. Качената информацията на самия уеб сайт е публикувана под лицензна CC0 1.0 Universal с цел да бъде безплатно употребявана като съдържание и да предоставя бъдещата възможност за колективно редактиране. Освен това, в съдържанието е включено детайлно описание на разработка и нейната инсталация, както и цялостно упътване за употреба в реална среда.

За да е пълноценно тестването на изградената софтуерна рамка, са изпълнени редица последователни проверки за качеството на разработката и за нейните възможности. Създаването на методология за тестване (Фигура 6) на рамката цели не само потвърждението на изградената функционалност в рамката, но и да се ускори бъдещата ѝ разработка в перспектива [Crispin`2008].

Точно поради тази причина, разделението на тестове в софтуерната рамка е на три основни части, които са независими една от друга като изпълнение (могат да се изпълняват паралелно):

- Тестване на функционалността;
- Тестване на съвместимостта;
- Тестване на качеството.



Фигура 6. Методология за тестване на рамката

Софтуерната рамка за криптографски услуги CryptoMañana притежава пълен комплект от изградени компонентни и интеграционни тестове, които са реализирани с помощта на софтуерната рамка PHPUnit. Крайният резултат за тестваемост на рамката (Фигура 7) е генериран като отчет от PHPUnit и потвърждава 100% покритие на компонентите ѝ.

	Code Coverage								
	Lines			Functions and Methods			Classes and Traits		
Total	[redacted]	100.00%	2405 / 2405	[redacted]	100.00%	431 / 431	[redacted]	100.00%	125 / 125
AsymmetricEncryption		100.00%	0 / 0		100.00%	0 / 0		100.00%	0 / 0
Compatibility	[redacted]	100.00%	344 / 344	[redacted]	100.00%	30 / 30	[redacted]	100.00%	5 / 5
Core	[redacted]	100.00%	1110 / 1110	[redacted]	100.00%	205 / 205	[redacted]	100.00%	60 / 60
CryptographicProtocol	[redacted]	100.00%	307 / 307	[redacted]	100.00%	53 / 53	[redacted]	100.00%	8 / 8
DataStructures	[redacted]	100.00%	40 / 40	[redacted]	100.00%	18 / 18	[redacted]	100.00%	6 / 6
Exceptions	[redacted]	100.00%	15 / 15	[redacted]	100.00%	15 / 15	[redacted]	100.00%	15 / 15
Factories	[redacted]	100.00%	125 / 125	[redacted]	100.00%	19 / 19	[redacted]	100.00%	5 / 5
Hashing	[redacted]	100.00%	180 / 180	[redacted]	100.00%	35 / 35	[redacted]	100.00%	17 / 17
Randomness	[redacted]	100.00%	163 / 163	[redacted]	100.00%	35 / 35	[redacted]	100.00%	3 / 3
SymmetricEncryption	[redacted]	100.00%	15 / 15	[redacted]	100.00%	4 / 4	[redacted]	100.00%	2 / 2
Utilities	[redacted]	100.00%	106 / 106	[redacted]	100.00%	17 / 17	[redacted]	100.00%	4 / 4

Фигура 7. Отчет за покритие на компонентните тестове

Софтуерната разработка е регистрирана под платформите Travis CI и Scrutinizer CI, които предоставят набор от услуги за непрекъснатата интеграция и доставка. Това позволява тестването за съвместимост, функционалност и качество да се извършва изцяло автоматизирано при публикуване на промяна в кода [Arachchi`2018]. Платформите автоматично изпълняват наличните тестове в хранилището на GitHub за всяка една от съвместимите версии на езика и публикуват доклад за покритието на кода. Изпълнението на компонентите тестове през платформата на Travis CI потвърждава 100% покритието на компонентите и съвместимостта на разработката, а Scrutinizer CI удостоверява качеството на кода и неговото форматиране, но и го оценява с максимална оценка 10.0 по собствената си комплексна скала за оценяване.

3.3 Експеримент

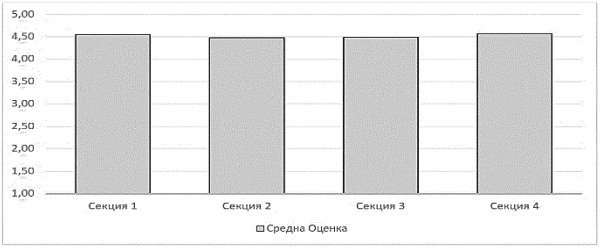
За да се оцени практическата приложимост на разработката е проведено изследване на възможността за интеграция на реализираната софтуерна рамка за криптографски услуги CryptoMañana и нейният криптографски модел. За да е възможно честното, справедливо и анонимно оценяване, създадената анкета е качена онлайн с помощта на услугата Google Forms.

Срокът на валидността на анкетата е 15 календарни дни, а направените въпроси в нея целят да са разбираеми за участници, присъствали на проведеното обучение по приложна криптография в ПУ, както и да обхващат по-сложни приложни теми от сферата на криптографията в

професионален аспект. Самият онлайн въпросник може да бъде достъпен чрез линка https://forms.gle/dAXfD62CkFdMcCFY9, който ще продължи да бъде наличен дори след преустановяването на гласуването, за да е възможно публичното изнасяне на резултатите от проучването. Въпросникът е създадени по начин, по който не е нужно да се отнема много време за попълване, но дава ясна оценка на отношението към интеграцията на софтуерната рамка за криптографски услуги и криптографския ѝ модел [Harris`2014].

За да се изяснят определените аспекти на качество на разработката и нейната приложимост, анкетата е разделена на четири основни секции. Общият брой на въпросите в нея е 24, а всяка от четирите секция съдържа по 6 запитвания. Целта им е да покажат до каква степен са съгласни анкетираните с твърденията във въпросника по Ликъртова скала (Likert scale) с 5 точкова система [Joshi`2015]. Анализът на резултатите от въпросника се базира на отговорите на анонимни потребители от публичното онлайн анкетиране. Общият брой на гласувалите е 47, а приемането на отговори се състоя между 30.08.2021 г. и 13.09.2021 г. Приемането на отговори е преустановено след 13.09.2021 г., а резултатите от анкетата са публично достъпни през същия линк и след тази дата.

Твърденията, дадени за секция „Практическа приложимост“ ясно показват, че потребителите имат положително мнение за практическата стойност на проекта, а средният резултат от оценяването е 4,54. За секцията „Архитектурен дизайн“ отново се наблюдава положително отношение като средния резултат тук е 4,47. Отчетените резултати при секция „Интегруемост, достъпност и ползваемост“ сочат към постигната интуитивност на компонентите в рамката и предоставят положителен среден резултат от 4,49. В последната секция „Криптографска функционалност“ се наблюдава най-висок резултат от 4,56, което е най-положително оценената секция и подсилва общото задоволство от публикувания софтуерен продукт. Наблюдава се обща позитивна тенденция по секция при визуализирането на средната оценка по секция (Фигура 8), което единствено подкрепя тезата за качеството на реализирания фреймуърк.



Фигура 8. Средно аритметично на резултатите от анкетата по секция

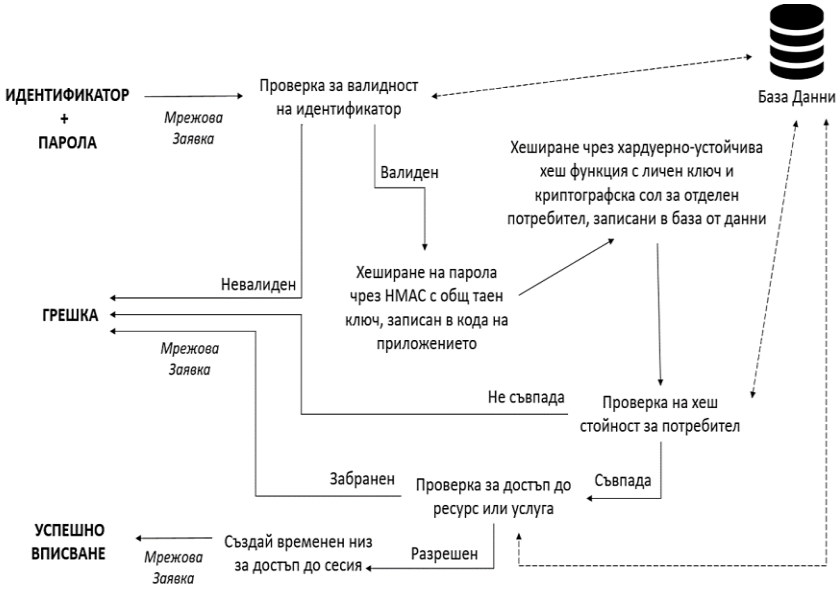
Отношението към софтуерната рамка за криптографски услуги CryptoManana е позитивно, като крайния среден резултат от анкетата е 4.52. Това означава, че мнозинството от попълнилите анкетата потребители смятат, че интегрирането на софтуерната разработка и нейната приложимост са от висока значимост в криптографски контекст на софтуерната сфера. Анализираните резултати затвърждават научната и приложна стойност на имплементирания криптографски модел и услуги в рамката.

ГЛАВА 4. МЕТОДИКА ЗА ИНТЕГРАЦИЯ НА КРИПТОГРАФСКИ УСЛУГИ

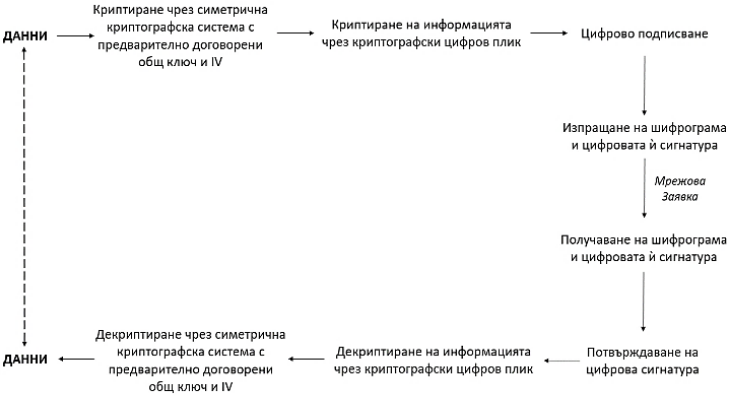
4.1 Осигуряване на защита по време на пренос

Осигуряването на защитата на данните по време на пренос и предотвратяването на подслушване от неоторизирани страни е задължително за всяка една съвременна мрежови свързана система или среда. В тази част на дисертационния труд са предложени методики за:

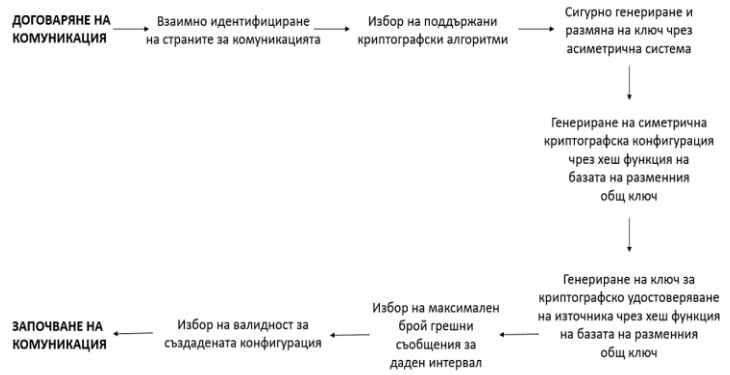
- Идентификация, автентикация и оторизация (Фигура 9);
- Сигурна мрежова комуникация (Фигура 10);
- Безопасна размяна на сесиен ключ (Фигура 11);
- Подход за „криптиране от край до край“ (Фигура 12).



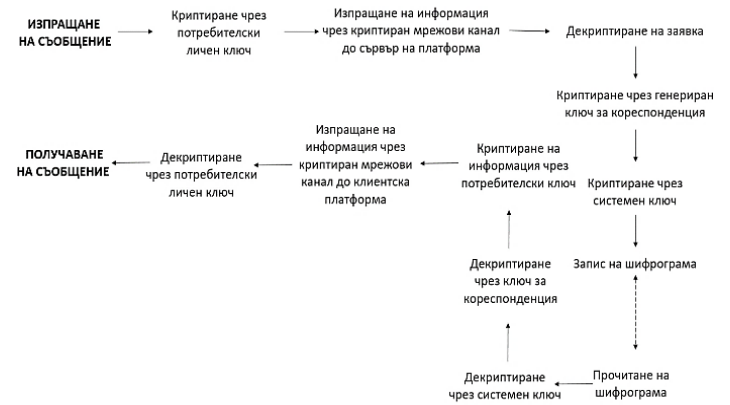
Фигура 9. Методика за идентификация, автентикация и оторизация



Фигура 10. Методика за сигурна мрежова комуникация



Фигура 11. Методика за безопасна размяна на сесиен ключ



Фигура 12. Методика за реализация на „криптиране от край до край“

Представените методики изискват стриктното спазване на изнесените принципи за сигурност и реализация, като предлагат и препоръчителни мерки за имплементация. Тези препоръки са в помощ на използващите методиката програмисти, проектанти и архитекти. Авторът извежда и конкретни примери за всяка една от методиките и установява директна връзка с криптографските услуги реализирани в софтуерната рамка CryptoMañana.

Предложените методики за имплементация на сигурна идентификация, автентикация и оторизация включват употребата на:

- Протокол за пароло-базирана автентикация;
- Протокол за симетрична автентикация;
- Протокол за асиметрична автентикация;
- Интеграция на двуфакторна или мултифакторна автентикация;
- Имплементация на механизъм за възстановяване на профил;
- Имплементация на механизъм за проследяване на история.

Проектираните методики за реализация на сигурна мрежова комуникация съдържат практическото прилагане на:

- Протокол за криптографски цифров плик;
- Интеграция на цифрови подписи;
- Интеграция на многопасово криптиране;
- Механизми за управление на активни сесии;
- Механизми за изтичане на конфигурация за връзка;
- Механизми за ограничаване на брой опити за свръзка.

Предложените методики за имплементация на безопасна размяна на сесиен ключ по несигурен канал включват употребата на:

- Протокол за размяна на ключове Diffie–Hellman;
- Прилагане на двойка от асиметрични системи;
- Прилагане на механизми за еднозначно взаимно идентифициране;
- Прилагане на механизми за трансформация на избран ключ;
- Задаване на валидност за избраната конфигурация;
- Задаване на максимален брой опити за конфигурация.

Проектираните методики за реализация на подхода за „криптиране от край до край“ съдържат практическото прилагане на:

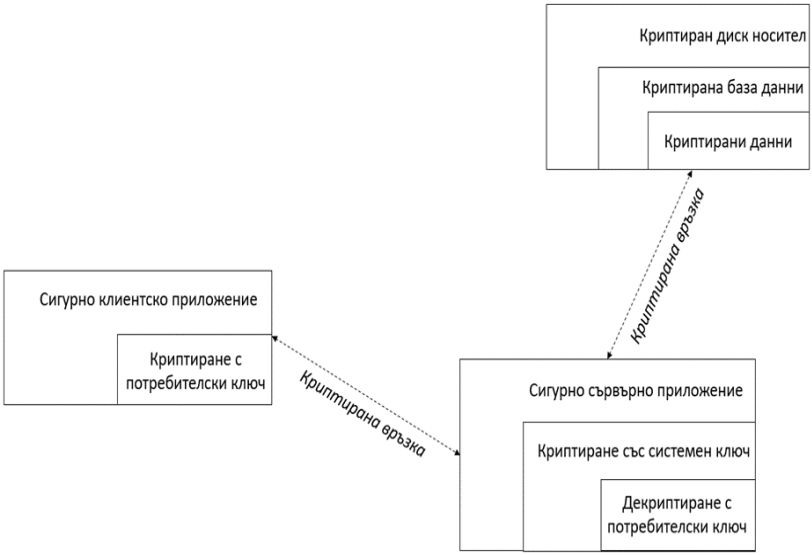
- Конфигуриране на криптиран трафик на мрежово ниво;
- Конфигуриране на криптиран трафик на приложно ниво;
- Криптиране на съхранението в база от данни или файлове;
- Криптиране на информацията на ниво дисково съхранение;
- Механизми за управление на множество от ключове за един или много потребители;
- Механизми за цифрово подписване на информацията.

Предложените методики са изцяло реализуеми чрез наличните функционалности на софтуерната рамка за криптографски услуги и се базират на най-добрите практики в киберсигурността. Създадените процеси за обработка на информация покриват изцяло нуждите на информационната сигурност и оползотворяват интегрираните криптографски системи [Mogollon`2008] [Moyle`2020] [Gupta`2018] [Bai`2020].

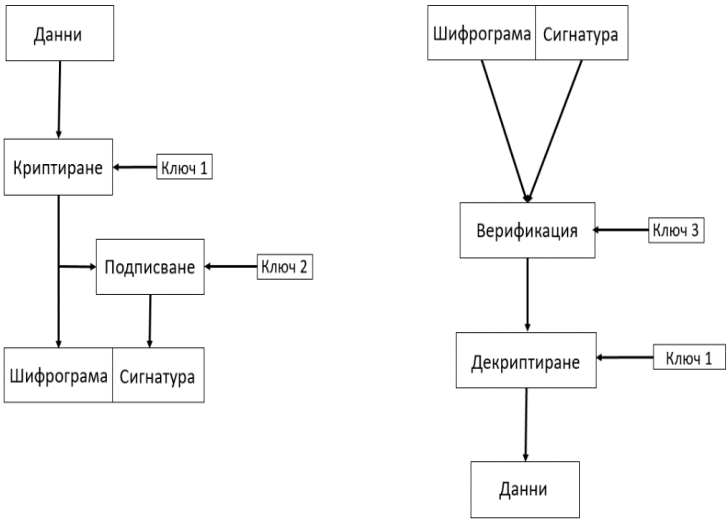
4.2 Осигуряване на защита по време на съхранение

Осигуряването на защитата на данните по време на съхранение и предотвратяването на неоторизиран достъп до тях е задължително за всяка една съвременна система, среда и платформа. Голяма част от съвременните софтуерни продукти са длъжни да спазват дадени регулации, за да им бъде легално позволено да функционират. Този тип изисквания обикновено включва сигурното съхранение на информацията под дадената платформа. В тази част на дисертационния труд са предложени методики за:

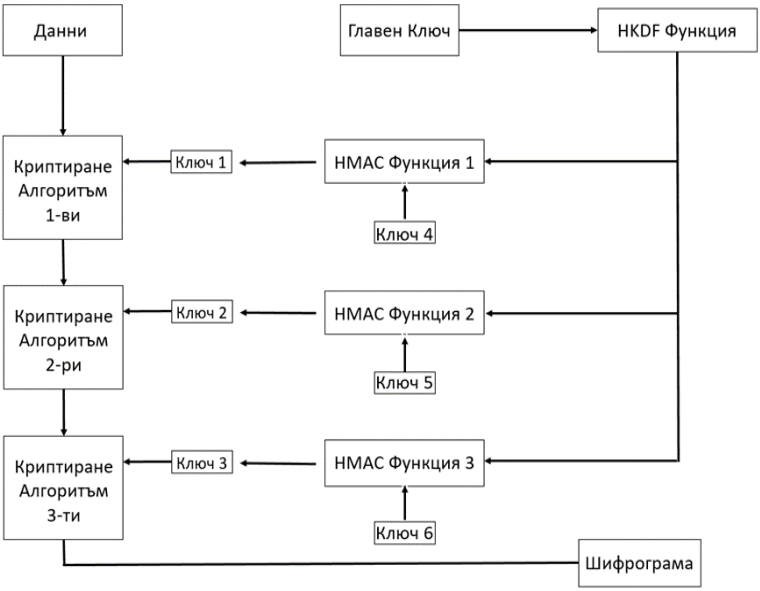
- Защитаване на физически устройства и бази данни (Фигура 13);
- Криптиране на данни с потвърждение за автентичността им (Фигура 14);
- Криптиране на лични и платежни данни (Фигура 15);
- Сигурно заличаване на информация (Фигура 16).



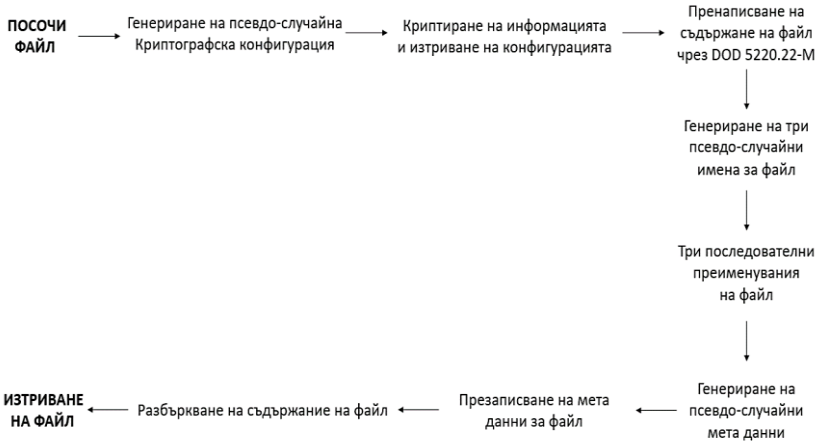
Фигура 13. Методика за защитаване на физически носители



Фигура 14. Методика за автентикирано криптиране



Фигура 15. Методика за криптиране на преизползваеми данни



Фигура 16. Методика за сигурно заличаване на данни

Представените методики задължават пълното покриване на изнесените принципи за качество и имплементация, като предлагат и допълнителни мерки за реализация. Тези препоръки са в помощ на използващите методиката програмисти, проектанти и архитекти. Авторът извежда и конкретни примери за всяка една от методиките и установява директна връзка с криптографските услуги предоставени от софтуерната рамка за криптографски услуги.

Предложените методики за имплементация на коректно защитаване на физически устройства и бази данни включват употребата на:

- Интегриране на пълно дисково криптиране;
- Интегриране на прозрачно криптиране на информацията;
- Интегриране на криптирани резервни копия;
- Механизми за криптиране на информацията на приложно ниво;
- Механизми за шифриране на временната памет;
- Създаване на отдалечени услуги за криптиране.

Проектираните методики за реализация на криптиране на данни с потвърждение за автентичността им съдържат практическото прилагане на:

- Протокол за автентикирано криптиране на данни;
- Проверка за идентичността, автентикацията и оторизацията на заявката;
- Механизми за проверка на източника на информацията;
- Механизми за проверка на целостта на данните;
- Механизми за цифрово подписване на информацията;
- Съхранение на допълнителна информация за източника на заявка.

Предложените методики за имплементация на сигурно криптиране на лични и платежни данни включват употребата на:

- Интеграция на симетрични криптографски системи;
- Интеграция на многопасово криптиране;
- Механизми за безопасно съхранение;
- Механизми за многослойна защита;
- Механизми за проследяване на промяна над данни;
- Механизми за проследяване на източника.

Проектираните методики за реализация на сигурно заличаване на информация съдържат практическото прилагане на:

- Интеграция на стандарти за заличаване от физически носител;
- Интеграция на многопасово криптиране преди изтриване;
- Механизми за премахване на информация от системни дневници;
- Механизми за премахване на информация от резервни копия;
- Механизми за премахване на информация от кеш или друга временна памет;
- Практики за периодично изтриване на информация с изтекла валидност.

Предложените методики са изцяло реализуеми чрез наличните функционалности на софтуерната рамка CryptoMañana и се базират на най-добрите практики в информационната сигурност. Дефинираните процеси за обработка на данните покриват изцяло изискванията за защита на данните и прилагат коректно криптографски услуги по време на реализацията си [Subangan`2019] [Aumasson`2017] [Grubbs`2017] [Reardon`2016].

4.3 Приложения

Приложенията включват подробна информация за използваните анкети и софтуерен код.

- Инсталиране на рамката чрез мениджър на пакети;
- Ръчно интегриране на рамката в процедурен код;
- Използване на външна библиотека за поддръжка на низове;
- Изключване на код за проверки за съвместимост с версията на езика;
- Изтегляне на код за документация и локално изпълнение на сайта;
- Генериране на техническа документация и отчет за тестове;
- Изпълнение на всички тестове за софтуерната рамка;
- Анкета за интеграция и приложимост на рамката на български език;
- Анкета за интеграция и приложимост на рамката на английски език.

ЗАКЛЮЧЕНИЕ

В разгледания дисертационен труд е показано, че осигуряването на защитата на данните по време на пренос и съхранение чрез криптографски услуги, става все по-важен и значим фактор при създаването на всички видове софтуерни продукти, както и представлява задължително изискване за голяма част от съществуващите регулации за обработка на информация. Успешно са създадени множество от методики за изграждане на цялостната защита на информационните системи и са предоставени съвети за практическата им реализация чрез дефинирани сигурни услуги, като част от изградения криптографски модел.

Създадената софтуерна рамка за криптографски услуги CryptoMañana и изграденият обектно-ориентиран криптографски модел дават възможност за практическа имплементация на дефинираните универсални методики за защитаване на информационни системи. В допълнение, те потвърждават използваемостта на проектирания криптографски модел, независимо от избрания език или платформа за реализацията му и успешно запълват липсата на такъв в езика за програмиране PHP.

АПРОБАЦИЯ НА ДИСЕРТАЦИОННИЯ ТРУД

Публикации

1. T. Karavasilev, S. Enkov. Object-Oriented Cryptography Hierarchy Realizations. IOSR Journal of Computer Engineering (IOSR-JCE), Volume 22, Issue 1, Ser. II (Jan - Feb 2020), 2020, pp 57-68. DOI: 10.9790/0661-2201025768, ISSN Online: 2278-0661, ISSN Print: 2278-8727.
2. T. Karavasilev. Object-Oriented Pseudo-Random Data Generator Realizations. IOP Conference Series Materials Science and Engineering, 618:012032, 2019, Scopus, Web of Science. SJR=0,198. ISSN: 1757-8981. DOI: 10.1088/1757-899X/618/1/012032.
3. T. Karavasilev, E. Somova. Overcoming the security issues of NoSQL databases. Journal of the Technical University - Sofia, Plovdiv branch, Bulgaria, Fundamental Sciences and Applications, Volume 24, 2018, pp 123-128. ISSN: 1310-8271.
4. T. Karavasilev, S. Enkov. Securing sensitive digital data techniques. IOSR Journal of Computer Engineering (IOSR-JCE), Volume 19, Issue 3, Ver. V (May-June 2017), 2017, pp 53-66. DOI: 10.9790/0661-1903055366, ISSN Print: 2278-8727, ISSN Online: 2278-0661.
5. S. Enkov, T. Karavasilev. Benchmarking hash functions. Journal of the Technical University - Sofia, Plovdiv branch, Bulgaria, Fundamental Sciences and Applications, Volume 23, 2017, pp 107-112. ISSN: 1310-8271.

Забелязани цитирания

1. D. Bulakh, A. Korshunov, S. Ilin. Identification of Integrated Circuits Based on Layout Layers Routing Information", 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus), 2021, pp 1965-1968. Scopus, Web of Science, SJR=2,383. DOI: 10.1109/ElConRus51938.2021.9396208.
2. W. Nwankwo. A Review of Critical Security Challenges in SQL-based and NoSQL Systems from 2010 to 2019. International Journal of Advanced Trends in Computer Science and Engineering, Volume 9, 2020, pp 2029-2035. Scopus, Web of Science. DOI: 10.30534/ijatcse/2020/174922020.
3. G. Wännberg, A. Forslöf. Behovet av NoSQL-databaser hos Business Intelligence-företag: Studie som undersöker behovet av alternativa databaser inom Business Intelligence-området. Informatik Student Paper Bachelor (INFSPB), Umeå University, 2020, pp 13. diva2: 1433627.
4. N. Chaudhry, M. Yousaf, M. Khan. Security assessment of data management systems for cyber physical system applications. Journal of Software: Evolution and Process, 32, 2019, pp 7. Scopus, Web of Science. DOI: 10.1002/smr.2241.

ОСНОВНИ ПРИНОСИ НА ДИСЕРТАЦИОННИЯ ТРУД

Основните приноси на дисертацията могат да се характеризират като **научни, научно-приложни и приложни**. С нейна помощ всеки един ново-разработван или вече съществуващ софтуерен продукт може да интегрира криптографски услуги и изгради адекватна цялостна защита на базата на най-ефективните практики в сферата на киберсигурността.

Проектантите и програмистите на софтуерни системи могат да използват създадените методики по време на разработването или поддръжката на своите продукти, за да покрият всички изисквания за нивото на сигурност, поставени от регулаторите или нуждите, дефинирани от потребители им.

Научните приноси са:

Н1. Създаден е общ, платформено независим криптографски модел за представяне на криптографски примитиви и услуги.

Научно-приложните приноси са:

НП1. Създадени са методики (набор от препоръки, правила, инструменти и съвети) за осигуряване на защита на информационни системи в различен приложен контекст;

НП2. Проектирана е софтуерна рамка за криптографски услуги на базата на анализ на съществуващи решения от различни технологични среди;

НП3. Предложена е унифицирана архитектура за управление на криптографски услуги и примитиви чрез платформено независим подход за реализация.

Приложните приноси са:

П1. Реализирана е софтуерна рамка за криптографски услуги като приложен инструмент за създаване, управление и конфигуриране на сигурни софтуерни услуги и среди;

П2. Интегриран е криптографски модел в език за програмиране без наличието на вградена поддръжка на криптографски услуги и примитиви;

П3. Извършено е тестване на създадената софтуерна рамка, автоматизирано и с реални потребители, и е направен анализ на получените резултати.

Моделите на методики и софтуерните средства, създадени в рамките на дисертацията, са използвани за създаване на учебни курсове в ПУ „Паисий Хилендарски“ (ПУ). Списък на проведени учебни курсове свързани с тематиката на дисертацията:

- Избираема дисциплина „Приложна криптография В Интернет с PHP“ през учебната 2017/2018 г.;
- Избираема дисциплина „Приложна криптография В Интернет с PHP“ през учебната 2018/2019 г.;
- Избираема дисциплина „Приложна криптография В Интернет с PHP“ през учебната 2019/2020 г.

БЛАГОДАРНОСТИ

С искрена благодарност и към всички колеги от катедра „Компютърна информатика“, за тяхната колегиалност, ентусиазъм и приятелско отношение. Искам да заявя личните си благодарности за подкрепата и добротата към проф. д-р Христо Крушков, който вече не е сред нас, поклон пред светлата му памет.

[Arachchi`2018] S. Arachchi, I. Perera. Continuous Integration and Continuous Delivery Pipeline Automation for Agile Software Project Management. 2018. DOI: 10.1109/MERCon.2018.8421965.

[Aumasson`2017] J. Aumasson. Serious Cryptography: A Practical Introduction to Modern Encryption. 2017. ISBN: 978-1593278267.

[Bai`2020] W. Bai, M. Pearson, P. Kelley, M. Mazurek. Improving Non-Experts' Understanding of End-to-End Encryption: An Exploratory Study. 2020. DOI: 10.1109/EuroSPW51379.2020.00036.

[Crispin`2008] L. Crispin, J. Gregory. Agile Testing: A Practical Guide for Testers and Agile Teams. 2008. ISBN: 978-0321534460.

[Goldreich`1998] O. Goldreich. Modern Cryptography, Probabilistic Proofs and Pseudorandomness (Algorithms and Combinatorics). 1998. ISBN: 978-3540647669.

[Grubbs`2017] P. Grubbs, T. Ristenpart, V. Shmatikov. Why Your Encrypted Database Is Not Secure. 2017. DOI: 10.1145/3102980.3103007.

[Gupta`2018] B. Gupta. Computer and Cyber Security: Principles, Algorithm, Applications, and Perspectives (Security, Privacy, and Trust in Mobile Communications). 2018. ISBN: 978-0815371335.

[Harris`2014] D. Harris. The Complete Guide to Writing Questionnaires: How to Get Better Information for Better Decisions. 2014. ISBN: 978-0615917672.

[Joshi`2015] A. Joshi, S. Kale, S. Chandel, D. Pal. Likert Scale: Explored and Explained. 2015. DOI: 10.9734/BJAST/2015/14975.

[Katz`2007] J. Katz, Y. Lindell. Introduction to Modern Cryptography: Principles and Protocols. 2007. ISBN: 978-1584885511.

[Menezes`1996] A. Menezes, P. van Oorschot, S. Vanstone. Handbook of Applied Cryptography. 1996. ISBN: 978-1439821916.

[Mogollon`2008] M. Mogollon. Cryptography and Security Services: Mechanisms and Applications. 2008. ISBN: 978-1599048376.

[Moyle`2020] E. Moyle, D. Kelley. Practical Cybersecurity Architecture: A guide to creating and implementing robust designs for cybersecurity architects. 2020. ISBN: 978-1838989927.

[Nelson`1995] N. Mark, J. Gailly. The Data Compression Book (2nd Edition). 1995. ISBN: 978-1558514348.

[Reardon`2016] J. Reardon. Secure Data Deletion (Information Security and Cryptography). 2016. ISBN: 978-3319287775.

[Schneire`2013] B. Schneire. Applied Cryptography: Protocols, Algorithms, and Source Code in C (2nd Edition). 1995. ISBN: 978-0471117094.

[Smart`2013] N. Smart. Cryptography: An Introduction (3rd Edition). 2013. ISBN: 978-0077099879.

[Subangan`2019] S. Subangan, V. Senthooran. Secure Authentication Mechanism for Resistance to Password Attacks. 2019. DOI: 10.1109/ICTer48817.2019.9023773.

[Vaudenay`2005] S. Vaudenay. A Classical Introduction to Cryptography: Applications for Communications Security. 2005. ISBN: 978-0387254647.

[Vogel`2011] O. Vogel, I. Arnold, A. Chughtai, T. Kehrer. Software Architecture: A Comprehensive Framework and Guide for Practitioners. 2011. DOI: 10.1007/978-3-642-19736-9.

[Wiegers`2013] K. Wiegers, J. Beatty. Software Requirements: Developer Best Practices (3rd Edition). 2013. ISBN: 978-0735679665.