МИНИСТЕРСТВО НА ОТБРАНАТА
ИНСТИТУТ ПО ОТБРАНА
„ПРОФЕСОР ЦВЕТАН ЛАЗАРОВ”

АВТОРЕФЕРАТ

на дисертационен труд

за придобиване на научна и образователна степен доктор

област на висшето образование:	5. Технически науки
професионално направление:	5.2 Електротехника, електроника и автоматика
докторска програма:	Автоматизирани системи за обработка на информация и Управление
първично звено:	научноизследователски отдел „Комуникационни и информационни системи и защита на информацията”
докторант:	**инж. Андрей Георгиев Иванов**
ръководител:	**полк. доц. д-р Николай Тодоров Стоянов**

София 2022 г.

Дисертационният труд съдържа 144 страници, от които 106 страници основна част и 21 страници приложение. В основната част на дисертацията са използвани: 10 фигури, 25 описани алгоритъма, 114 литературни източници и научни публикации.

Номерирането на главите, фигурите, таблиците, формулите и цитираната литература в автореферата съответства на това в дисертационния труд. Номерацията на използваната литература в автореферата съответства на номерацията на използваната литература в дисертационния труд.

Защитата на дисертационния труд ще се проведе на _____ от _____ часа в зала _____ на Институт по отбрана „Професор Цветан Лазаров“, на открито заседание на научно жури в състав:

1. _____
2. _____
3. _____
4. _____
5. _____

Резервни членове:

1. _____
2. _____

Научен ръководител:

полк. доц. д-р Николай Тодоров Стоянов

Автор:

инж. Андрей Георгиев Иванов

Заглавие:

Анализ на устойчивостта на криптографски системи с публичен ключ

ВЪВЕДЕНИЕ

През вековете на своето съществуване човечеството се отличава от останалите животински организми на планетата със способността си да усъвършенства някои специфични свои умения. Те са част от всички сфери на неговия живот и дейност. Човек създава и развива своите инструменти на труда, начин на мислене и комуникация. В началото човешките индивиди са използвали основно речеви методи за комуникация и сравнително обикновени, немногоцветни картини. С развитие на уменията и познанията си хората създават по-добри и усъвършенствани механизми за натрупване на информация и комуникация помежду си. Така в нашето съвремие ежедневно използваме електронни средства за комуникация, съхранение и обработка на информация.

През вековете човек е осъзнал, че информацията, която притежава, нейното надеждно съхранение и опазване, има изключително важна роля за успеха на неговото индивидуално и социално предимство в сравнение с останалите индивиди. Така още от столетия са създавани и се създават средства, даващи възможност за надеждна и сигурна защита на информация с различен обем и структура. Най-широко използваният способ за защита на информация е нейното криптиране. Криптографията (от гръцки криптос-„скрит“ и графо-„пиша“) изучава принципите, средствата и методите за преобразуване на данни, за да бъде прикрита тяхната семантика или да бъдат защитени от несанкциониран достъп. Тя е и средство, използвано с цел защита срещу промяна на съдържанието или структурата на информацията от лица, чийто достъп до реалното ѝ съдържание, трябва да бъде ограничен.

Това се постига чрез математически алгоритъм, чрез който данните претърпяват промяна в зависимост от най-малко един таен параметър, наричан „ключ“ (криптографски ключ), известен само на участниците в комуникацията или хората, на които е делегиран достъп до нея.

Развитието на технологиите и най-вече на електрониката в последното столетие предизвиква необходимостта от използване на криптография до ниво - човешки индивид. Така в наши дни в своите ежедневни дейности хората използват технически електронни средства, в които реализацията на криптографски алгоритми и способи за защита на информацията е задължително и вече се приема за естествено тези средства за защита да бъдат използвани в почти всички сфери на човешката дейност.

В зависимост от математическите си основи и схема на използване, в криптографията са обусловени две основни групи алгоритми. Тяхното условно разделяне е в зависимост от размера, характеристиките на ключа и начина на неговото използване при защитата на данни. Тези две групи алгоритми са:

- **„Криптографски алгоритъм със симетричен ключ“**;
- **„Криптографски алгоритъм с публичен ключ“**.

Широкото използване на публичната криптография в ежедневието на съвременния човек, с цел сигурна и надеждна защита на данни от личен, обществен, социален и държавен характер, я превръща в значим фактор за функционирането на едно общество, отделните държави и света като цяло.

Всичко изложено до момента доказва голямата значимост на криптографските системи за защита на информацията, чрез използването на публичен ключ. Разработката на дисертационния труд е мотивирана от съществуващия въпрос, отнасящ се до определяне на степента на устойчивост при функционирането на криптографски системи с публичен ключ.

Свидетелства за наличието на уязвимости в реализирани и използвани криптографски системи с публичен ключ все по-често се публикуват в интернет пространството. Тук е уместен въпросът колко от реално установените уязвимости се оповестяват официално, за да бъдат отстранени. С нарастването на информационните технологии, големи компании, държавни организации и хакерски групи (които могат да се окачествят като **недоброжелатели**) целят да си осигурят по-лесен несанкциониран начин за достъп до информационни масиви, които са чужда собственост. Най-незабележимият и лесен начин това да бъде реализирано, е чрез компрометиране на криптографските системи. Недоброжелателите се стремят да постигнат това, използвайки предварително създадени задни врати (алгоритмични, математически или др.) за достъп до личните ключове (използвани в публичната криптография), което може да доведе до разкриване на явната информация (първообразът на защитаваната информация).

Тези причини за съществуване на възможност за несанкциониран достъп за възстановяване на частните ключове на криптографска система с публичен ключ е предпоставка да бъдат оценени техните математически основи и принцип на функциониране. Необходимо е да се обобщят и анализират съществуващи хардуерни средства и софтуерни библиотеки, използвани за криптографски цели в практиката. Трябва да се анализират и оценят показателите и параметрите на изследваните криптографски средства и на база резултатите да се оцени съществува ли скрита възможност, те да бъдат компрометирани чрез бързо и лесно възстановяване на частния ключ, което оказва пряко съществено влияние върху устойчивостта на функциониране на криптографските системи, използващи публична криптография.

Предмет на настоящата работа е да се разгледат и анализират най-широко прилаганите в практиката алгоритми и подходи, използвани в публичната криптография с цел защита на информация. Да се оцени степента на тяхната устойчивост на атаки, свързани с наличието на слабости, които могат да съществуват в следствие на преднамерено създадени уязвимости в основните криптографските примитиви на съответния алгоритъм. Да се разработят модели, които да послужат като основа за доказване на вероятна възможност за съществуването или не съществуването на уязвимости от такъв вид.

Дисертационният труд е структуриран: увод, три глави, заключение, литература, списък на публикации, свързани с дисертацията, списък на използваните термини, математически означения и основни функции, списък на фигурите.

В първа глава са разгледани математически и алгоритмични основи на публичната криптография, описана е тяхната приложност и сфери на използване, систематизирани са прилаганите в практиката стандарти и изисквания към системи, използващи публичната криптография. Направена е оценка на

проблемите, свързани с устойчивостта на функциониране на криптографски системи с публична криптография и са формулирани: целта и задачите при реализация на разработвания научно-изследователски труд.

Във втора глава са предложени два модела на решение. Първият е реализиран с цел постигане на по-голяма ефективност и достоверност на резултата за определяне на делимост на число, чрез използване на вероятностния алгоритъм на Милър-Рабин. Вторият модел на решение представя математическата възможност за по-ефективно изпълнение на алгоритъма на Силвър-Похлиг-Хелман, чрез който се реализират атаки към най-разпространения в практиката към настоящия момент криптографски алгоритъм в публичната криптография, RSA.

В трета глава са описани математическите основи и модел на нов подход за реализация на клептографски алгоритъм. Описано е: неговата практическа реализация, генериране на домейни за публичен ключ, извършване на оценка на техническите показатели на реализацията му и е направен сравнителен анализ между предлагания алгоритъм и други съществуващи клептографски алгоритми. В тази глава е направен анализ и оценка за възможността за съществуване и използване на клептография в практико-приложни системи, базирани на публична криптография. Направени са изводи и е предложен механизъм, чрез който да се избегнат уязвимости, основаващи се до клептографски атаки към системи, базирани на RSA.

В заключението са описани приносите на дисертационния труд, както и насоки за бъдеща работа.

За изпълнение на основните задачи в дисертационния труд са приети следните ограничения:

- Разгледан и анализиран за устойчивост е единствено алгоритъм RSA, използван в публичната криптография.
- Разгледани, анализирани и оценени са единствено възможности за влияние върху устойчивостта на криптографски системи с публичен ключ, свързани с генерирането на RSA ключ и възможностите на атаки за неговото разбиване.

ПЪРВА ГЛАВА

ПУБЛИЧНА КРИПТОГРАФИЯ, ОСНОВИ, АЛГОРИТМИ, ПРИЛОЖЕНИЕ И СФЕРИ НА ИЗПОЛЗВАНЕ. УСТОЙЧИВОСТ НА ФУНКЦИОНИРАНЕ.

Средство за защита на информацията, което се основава на изключително трудно решими математически задачи, чиято трудност остава висока дори при наличие на бързо функциониращи и добре организирана в огромни клъстери съвременна изчислителна техника е публичната криптография. За постигане на тази цел в реализацията на криптографските алгоритми, попадащи в този дял на криптографията се използват математически операции, които са част от раздела „Теория на числата“ [8, 41].

1.1 Фундамент на публичната криптография. Алгоритми за криптиране. Сфери на използване. Устойчивост при функциониране.

База за създаване на алгоритми в публичната криптография са следните трудно решими математически задачи: „Разлагане на големи числа на прости множители“ (Number Factorization), „Изчисляване на дискретен логаритъм в крайни полета“ (Discrete Logarithm in Finite Field - DLP) и „Изчисляване на дискретен логаритъм на елиптична крива“ (Elliptic Curve Discrete Logarithm Problem - ECDLP). Математическите примитиви, използвани за целите на публичната криптография са свързани с изчисления и операции в математически конструкции от тип мултипликативни и адитивни абелиеви групи [74, 95].

1.1.1 Фундамент на публичната криптография. Алгоритми за криптиране.

Изчисляване на дискретен логаритъм в крайни полета е задача, при която ако a е примитивен елемент (генератор) на крайно поле F_p и d е произволен елемент на F_p $(d \in F_p)$, е трудно да се изчисли x при известни d, a и p, за които: $d = a^x \ mod \ p$. Такава задача е сложна за изчисление, защото съществува периодичност и са на лице голямо количество стойности на $x \in Z$, за които е изпълнено равенството: $d = a^x \ mod \ p$. Това е вярно вследствие от факта, че равенството: $a^x = a^{x+k(p-1)} \ mod \ p$ е изпълнено за всички $k \in Z$.

В тази подточка на дисертационния труд са представени следните алгоритми:

Алгоритъм на Дифи-Хелман (Diffie-Helman) …
Алгоритъм на ElGamal …
RSA …
DSA …
Elliptic Curve Cryptography (ECC) …

1.1.2 Сфери на използване на публичната криптография.

Най-голям тласък за навлизане на публичната криптография оказва развитието на технологиите, използвани в Интернет среда. Това е логично и се дължи на факта, че все по-често се налага информация по защитен способ да бъде обменяна между страни в комуникацията, които са практически непознати една на друга. Защита, постигаща търсени нива на сигурност се реализира, чрез използването на комбиниран подход, използващ публичната криптография за цифрово подписване или защита на криптографски ключове, използвани при симетрично криптиране.

Съхранението на криптографски ключове, използвани в публичната криптография се постига, чрез изграждането на подходяща организация и инфраструктура. Такава форма на организация е наречена „Инфраструктура на публичните ключове“ (Public Key Infrastructure - PKI) [100, 101, 112]. В нея ключовете се съхраняват под формата на цифрови сертификати.

Целта на PKI е да улесни защитения електронен трансфер на информация за редица мрежови дейности като електронна търговия, интернет банкиране, поверителен имейл и много други. PKI се изисква за дейности, при които паролите са неподходящ метод за удостоверяване и се търси по-строго доказателство за потвърждаване на самоличността на страните, участващи в комуникацията и за валидиране на прехвърляната информация.

В криптографията PKI е споразумение, което свързва публичните ключове със съответните самоличности на обекти (като хора и организации) и служи за надеждно и сигурно съхраняване на издадените цифрови сертификати. Обвързването се установява чрез процес на регистрация и издаване на сертификати от сертифициращ орган (Certificate Authority - CA). Всеки регистрационен орган е отговорен за една или повече от следните функции:

- идентифициране и удостоверяване на кандидатите за сертификат;
- одобрение или отхвърляне на заявления за сертификат;
- иницииране на анулиране или спиране на сертификати при определени обстоятелства;
- обработка на заявки на абонати за отмяна или спиране на техните сертификати;
- одобряване или отхвърляне на заявки от абонати, за да подновят своите сертификати.

Съдържанието и структурата на един цифров сертификат се стандартизира с прилагането на X.509 (стандарт на „International Telecommunication Union“ - ITU). Всеки сертификат трябва да бъде подписан само от една страна (от CA). Всяка CA притежава свой собствен основен сертификат (коренен сертификат). Може да съществува йерархична подчиненост между отделни CA.

Един от най-разпространените случаи на използване на цифров сертификат е удостоверяването на електронна страница. Този принцип е възникнал като средство на противодействие на измами в интернет пространството, реализирани чрез фиктивни интернет страници, създавани с цел събиране на лична или чувствителна информация.

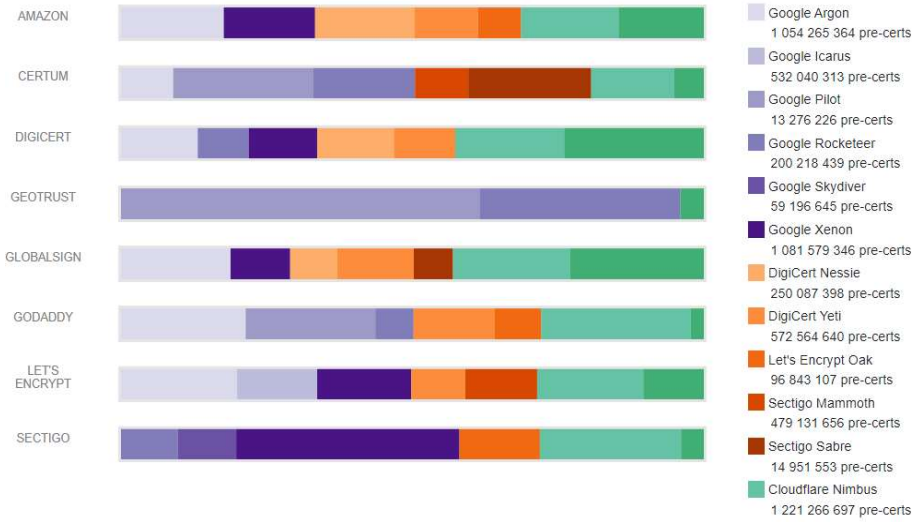
С цел осигуряване на най-добра сигурност в тази област, част от CA са създали и са се обединили в организация „Прозрачност на Цифрови Сертификати“ (Certificate Transparency organization - CTo) [87, 102, 114]. CTo внася прозрачност в система, използваща SSL/TLS, която поддържа мрежата. Протоколите SSL/TLS реализират криптографски операции чрез прилагането на публична криптография

Използвайки статистика на CTo (фигура 1) можем да придобием реална представа за количеството сертификати, които се използват в интернет пространството.

Към декември 2021г. това са над стотици милиони сертификати. Показателни са следните две цифри от глобалната статистика:

- Брой на ново генерирани сертификати – 256 788 certs/hr
- Брой на изтекли сертификати - 239 840 certs/hr

Разликата в количеството е 16 948 броя сертификати на час в полза на новоиздадените [7, 89].



Фиг. 1

Количество цифрови сертификати в реална употреба

https://ct.cloudflare.com/

Един друг аспект на сигурност в интернет пространството е използване на цифрови сертификати за подписване на софтуерен код. Друга важна употреба е безопасното предоставяне на актуализации и корекции на съществуващ софтуер [17]. Операционни системи Windows, Mac OS X и повечето дистрибуции на Linux предоставят актуализации, чрез подписване на код, за да се гарантира, че не е възможно трета страна, злонамерено да разпространяват код, с което да повлияе на сигурността на различни електронни системи. Цифрово подписаният код е средство за проверка, дори ако е доставен от трета страна.

Удостоверяване, базирано на цифров сертификат, е отговорът за предотвратяване на измами при комуникация на ниво машина-машина. Следователно, организации като банки и други финансови институции могат да комуникират помежду си вътрешно и да изграждат доверени комуникации между тях и това да гарантира, че нито една измамна система или злонамерен софтуер не може да поеме контрол върху тяхна или обща споделена инфраструктура.

Нараства броя на правителствата по света, които въвеждат електронни системи за регистрация и цифрова идентичност на техните граждани в националните идентификационни системи. Причини за това са: улесняване на достъпа до услуги, национална сигурност, борбата срещу корупцията и други.

Взаимното признаване на цифровите подписи в Европейския съюз позволява сключването на споразумения и търговски сделки между правителства и/или между компании. Регламент за електронна идентификация и доверени услуги (Regulation of Electronic Identification and Trusted Services – eIDAS 910/2014 / ЕС) опростява и стандартизира идентификационните данни: цифрови самоличности и цифрови и електронни подписи. Той създава „единен цифров пазар“, за да гарантира сигурни цифрови транзакции между членовете на Европейския съюз (ЕС) и страните, с които имат международни отношения.

В обобщение на сферите на приложение и използване на публичната криптография можем да изброим: сигурно и безопасно комуникиране в интернет среда, защита на персонална идентичност, сигурност на използвания

софтуер, сигурност във все по разпростиращата се по обем комуникация машина-машина, цифрова идентичност на населението и много други.

Към декември 2021г. процента на приложение на RSA е приблизително 93%. Този резултат е на база статистика, реализирана чрез проверка на коренните сертификати в дистрибуциите на основни операционни системи (Windows, Linux/Android и MAC OS), както и предоставена статистика от CTo показан на фигура 2.

Фиг. 2

Съотношение между използваните цифрови сертификати, базирани на RSA и ECDSA

https://ct.cloudflare.com/

1.1.3 Ключове, използвани в публичната криптография. Генериране, стандарти и рискове за устойчивостта.

За всяка криптографска система един от най-важните елементи на сигурността е ключът на потребителя, който се използва за криптиране на информация. Ето защо процесът на генериране на ключове винаги е важен етап при защита на данни. Влиянието, което този процес оказва върху сигурността и устойчивостта на криптографски системи с публичен ключ е огромно. Въпросите, свързани с изследване на устойчивост на криптографски системи, прилагащи публична криптография са свързани с процесите на генериране и съхранение на използваните частни ключове на потребителите [10, 24, 30, 31, 37, 67]. Ако една криптографска система допуска генерирането и използването на криптографски ключове, които могат лесно да бъдат компрометирани, се оценява като неустойчива [75, 76, 78].

Големият обхват на използване на RSA в практиката, подчертано като резултат в предходната подточка, е причината процеса на изследване на устойчивостта на криптографски системи с публичен ключ в дисертационния труд да бъде насочен към оценка на устойчивостта на работа на RSA.

Важно значение за правилното и устойчиво функциониране на криптографските системи, базирани на публична криптография има прилагането на стандарти в процесите, свързани с генериране на ключове [92, 110]. Генериране на ключове за RSA криптографски системи се регулира от стандарти, които се прилагат в процеса на създаване на двете части на ключа [84, 90, 96].

Най-широко прилагани в практиката са два стандарта, разработени от „Националния Институт по Стандартизация“ на САЩ (NIST) и „Европейския Институт по Телекомуникационни Стандарти“ (ETSI), в които са посочени поредица

от стъпки и условия, на които трябва да отговарят числата съставляващи домейните на ключ, използван в криптографски системи, използващи RSA.

В практиката генерирането на ключовете се реализира по софтуерен начин, чрез приложения за компютърна конфигурация или в специализиран хардуерен модул, който притежава интерфейс за комуникация с него. Проверките за покриване на посочените по-горе стандарти се извършват в специализирани лаборатории и всяко средство за генериране на ключове притежава период на валидност за използване.

Въпреки изискванията на стандартите и строгия контрол върху работата на хардуерните устройства и софтуерните приложения, използвани за криптиране, съществува риск процеса на генериране на ключ да е компрометиран.

В криптографски системи, базирани на RSA гарантирането, че p и q са избрани с достатъчна произволност, е решаващ компонент за запазване на сигурността на публичния ключ. Разлагането на голям модул N за получаване на p и q не е възможно при нормални обстоятелства. Въпреки това, ако ключовете се генерират с лоша произволност (не напълно произволни множители, генерирани по напълно случаен принцип), тогава е възможно да се допусне, че два RSA публични ключа ще споделят множител и ако се генерират достатъчно ключове е възможно получените модулни числа N_i да бъдат разложени чрез търсене на техен общ делител ($GCD(N_i, N_j)\ i \neq j\ \ N_i \neq N_j$) [46, 108].

През 2012 г. група изследователи с участието на Ленстра [46] реализират изследване, в което анализират приблизително 6,2 милиона цифрови сертификати, които са практически използвани в интернет пространството и откриват, че около 4,3% от тях притежават споделени множители. При анализ на 45 милиона модулни числа, елемент на RSA ключове, сканирани в периода 2015-2017 г. (от реално функциониращи цифрови сертификати), използвайки първия милион малки прости числа, са разбити 192 709 ключа. Това количество съответстват на 344 055 различни сертификата в оригиналния набор от данни, или 0.56%.

Тази статистика и анализ показва реален проблем в принципа и подхода за генериране на множители за модулни числа използвани в сертификати, предназначени за RSA криптографски системи.

Един пример за проблем при стандартизирано и проверено като функционалност хардуерно средство или софтуерен продукт, е в „доверителен модул за сигурност“ (Trusted Platform Module - TPM), произвеждан от компания Infineon. Проблемът в чиповете на Infineon е установен в процеса на експлоатация, а не при сертифициране. Модули от тип TPM са електронни чипове, в които се генерират и съхраняват ключове за криптографски алгоритми. В конкретния пример с чиповете произведени от Infineon, са засегнати чипове с версии на софтуера: 4.0 - 4.33; 4.4 - 4.42; 5.0 - 5.61; 6.0 - 6.42; 7.0 - 7.61; 133.0 - 133.32; 149.0 - 149.32. Те са основно използвани в компютърни конфигурации и засягат сигурността на операционни системи Windows и Linux [93, 99]. Най-важното в примера е установеното проявление на уязвимостта: ***„Уязвимостта позволява възстановяване на личен ключ, когато притежавате само публичен ключ“***. В публикацията на Националната Агенция за Сигурност на САЩ (NSA) се казва, че

тази уязвимост засяга в значителна степен устройства, използвани от министерство на отбраната (Department of Defense - DoD) [81, 88, 109].

Съществува и друга група подходи за атака срещу RSA базирани криптографски системи, които са най-трудни за установяване и е трудно тя да се причисли към някоя от другите класически категории на атаки. Това са атаки, дължащи се на предварително заложени математически конструкции, които единствено в определена комбинация между математическия модел на алгоритъма и генериран ключ създават уязвимост [40, 58]. Групата на такива видове способи за атака е наименувана ***„Клептографски способи за атака“***. Целта на **клептографията** е да се създадат предпоставки за атака, чрез които да е възможно реализация на уязвимост само от запознати с начина на функциониране на слабостта в математическите и криптографските примитиви на алгоритъма. За първи път терминът клептография е предложен от Адам Йоунг (Adam Young) и Моти Юнг (Moti Yung) през 1996 година [3, 18, 19, 26]. Една от статии си Юнг и Йоунг озаглавяват: „Kleptography: Using Cryptography Against Cryptography“ [20, 25, 39, 77].

Към настоящият момент липсва строга класификацията на клептографските атаки. Това се дължи на факта, че няма ясен формален модел на клептографските механизми, и това че се използва широк спектър от атаки срещу тях. Общото описание на клептографията е: способ, включващ методи за изграждане на канали за тайно изтичане на чувствителна криптографска информация или разработване на криптопримитиви с частично нарушаване на криптографски свойства при определени условия.

Един от ярките примери, станали достояние на световната общност през 2021г. е тайна договорка между САЩ и Германия, започнала след края на Втората световна война. Страните са се договорили и придобили собствеността върху швейцарска компания (Crypto AG), разработваща и комерсиално предлагаща криптографски средства на правителства в цял свят (над 60 държави). Част от функционалността на криптографските средства на Crypto AG, позволява на запознатите страни бързо да извличат криптографските ключове и така да получават достъп до явните данни [80].

В тази част на дисертационния труд е направено представяне на алгоритъма, предложен от Йоунг и Юнг.

Клептографска атака на Йоунг и Юнг срещу RSA ...

1.2 Цели и задачи на дисертационния труд

Широкото използване на криптографски алгоритъм RSA, както и наличието на предпоставки и примери, доказващи потенциална възможност за наличие на рискове, засягащи устойчивостта на неговото функциониране, насочи разработката на дисертационния труд в търсене на потенциални възможности за създаване на модели и решения, които да засягат неговата устойчивост и сигурността на криптираните. Наличието на модели и решения, отслабващи или засилващи устойчивостта на RSA ще повлияят в най-голяма степен сигурността на криптираните данни, които се обменят ежедневно от милиони кореспонденти.

В следствие на направените анализи формираната цел при разработката на дисертационния труд е следната: да се предложат модели на решение, позволяващи проверка на съществуващи обективни възможности, оказващи

влияние върху устойчивостта на функциониране на криптографски системи, базирани на криптографски алгоритъм RSA.

За постигане на поставената цел е необходимо да се решат следните задачи:

1) Да се анализират алгоритмите, използвани за оценка на делимост на число, като се изследват принципите на работа на най-разпространените алгоритми в тази област. Да се оценят техните слабости. Да се анализира възможността за съществуване на решения, които да доведат до повишаване на ефективност и достоверност на получените резултати и ако е възможно да се постигне повишаване на бързодействието в процеса на оценка за делимост;
2) Да се анализират съществуващите алгоритми за атака на RSA, като се оцени възможността за създаване на формален модел, който да позволи повишаване на ефективността за разлагане на числа на прости множители;
3) Изследвайки математическите конструкции на алгоритмите за криптиране и тези за тяхната атака да се потвърди или отхвърли възможността за съществуване на реална заплаха за устойчивостта на функциониране на RSA, чрез създаване на слабост в процеса на генериране на личния и публичния ключ.
4) Да се създаде функционален модел на практическа софтуерна реализация, на основата на която да се извърши анализ на ефективността на резултатите.

1.3 Изводи.

- Прилагането на алгоритми за намиране на решения на задачи в областта на теория на числата повишава сложността за реализация на атаки спрямо криптографските примитиви на алгоритми, използвани в публичната криптография.
- Най-разпространения в практиката алгоритъм за криптиране на данни, чрез използване на криптография с публичен ключ е RSA. Наличие на уязвимост, която да компрометира устойчивостта на работа на криптографска система, използваща RSA, би застрашило сигурността на голям брой комуникационни и информационни системи и по този начин би рефлектирало негативно и в големи мащаби върху големи социални групи като цяло.
- Два са основните подходи, даващи реална възможност за постигане на ефективна атака спрямо RSA. Първият е свързан с принципите и начините за създаване на криптографски ключове. Това включва степента на достоверност на резултата, който алгоритмите за оценка на делимост на число предоставят. Втория е свързан с възможността за наличие на скрити (от потребителя на RSA) механизми, които да позволят лесно компрометиране на личния ключ, с което да се окаже негативно въздействие върху устойчивостта на функциониране на RSA.
- Основавайки се на всичко изложено до момента, включващо: теорията на която се основават алгоритмите за криптиране с публичен ключ; начините на

тяхното използване в практиката; принципите и способите за генериране и оценка на прости числа като част от ключове, предназначени за криптиране в публичната криптография, може да се направи заключението, че съществуват рискове за устойчивото функциониране на публичната криптография, което да повлияе на сигурността на криптираните данни.

- Технологичното развитие на електронните средства не е единствената предпоставка за създаване на рискове за устойчивото функциониране на криптографски системи, използващи RSA. Потенциалното съществуване на възможност за внедряване на клептографски криптомеханизми сериозно може да окаже негативно влияние във всички сфери, в които се използват алгоритми за криптиране, базирани на проблем, свързан с разлагане на големи числа на прости множители.

ВТОРА ГЛАВА

МОДЕЛИ НА РЕШЕНИЯ, ОКАЗВАЩИ ВЛИЯНИЕ ВЪРХУ УСТОЙЧИВОСТТА НА RSA.

В тази глава са разгледани два варианта на модели на решения, засягащи устойчивостта на криптографски системи, използващи алгоритми за криптиране с публичен ключ. Първият е свързан със създаването на модел за оценка делимост на числа, целящ подобряване на резултатите при използване на алгоритъма на Милър-Рабин. Вторият модел на решение реализира идеята за постигане на по-ефективна атака на RSA базирани криптографски системи, чрез използването на алгоритъма за атака, предложен от Силвър-Похлиг-Хелман (Silver-Pohlig-Hellman) [83, 86].

2.1 Модел на решение, позволяващ подобряване на оценката за делимост на число при използване на алгоритъма на Милър-Рабин

В публичната криптография от особена важност за постигане на висока сигурност и надеждна защита на криптираната информация са простите числа. Те са основен градивен елемент на частите (домейните) на всеки публичен ключ, използван при реализацията на алгоритъм RSA. Това е причината в тази част на дисертационният труд да са разгледани някой важни алгоритмични аспекти в областта на делимост на числата („Primality Tests“). Това е направено с цел постигането на по-голяма яснота на връзката между предлагания модел, чиято цел е да послужи като допълнение към един от най-широко използваните в практиката алгоритми, този на Милър-Рабин (Miller-Rabin).

Способността за достоверно определяне делимост на едно число има голямо и важно значение за сигурността на криптираните данни и устойчивостта на функционирането на криптографски системи за криптиране, използващи публичен ключ [47, 48, 49]. За постигане на целите на криптографията, размера на простите числа, които се използват е много голям, в порядъка на стотици цифри в десетична бройна система, което е еквивалентно на хиляди битове в двоична бройна система.

Определяне делимост на число е въпрос, занимавал математиците още от античността. За постигане на тази цел в началото са правени опити да бъдат синтезирани и изведени формули за изчисляване на просто число като функция

от входни параметри, но с развитие на теорията на числата е установено, че не съществува еднозначен математически апарат от функции, който детерминистично да важи за всички съществуващи прости числа [44, 63]. Вследствие на този факт, единственият възможен подход е прилагането на алгоритъм за проверка дали число е делимо на съставни множители или не (множители различни от 1 и самото число).

Практически в тази област съществуват две основни групи алгоритми: детерминистични и вероятностни. В коя от двете групи попада даден алгоритъм за определяне делимост на число, зависи от това с каква степен на определеност е отговора.

Вероятностни алгоритми

Създадените през годините вероятностни алгоритми са не малко на брой и всеки един от тях се характеризира с бързина при оценка на делимост многократно по-голяма, от който и да е детерминистичен алгоритъм. Вероятността едно число да се оцени като съставно, ако то реално е просто няма да повлияе на сигурността на защитаваните данни, защото то би било пропуснато и за участие в ключ. За съжаление в тази група алгоритми характерната грешка е в това, че съставни числа се определят като прости. Ако едно голямо число p с размерност от няколко хиляди бита се оцени като просто, а то реално е съставно, ще се допусне грешно предположение за размерността на полето F_p, в което се реализирала цялата аритметика, прилагана в процеса на криптиране [9]. Това от своя страна създава предпоставка, ако недоброжелател установи делимост на p и разложи това число на прости множители, той да реализира успешна атака срещу RSA.

Най-разпространен за използване вероятностен алгоритъм във всички съвременни системи, реализиращи публична криптография е този на „Милър-Рабин алгоритъм за делимост“ („Miller-Rabin Primality Test“) [82, 97].

Алгоритъмът на Милър-Рабин се базира на трети подход, Да се открие число x за което да са изпълнени:

$$x^2 \equiv 1 \; mod \; p \text{ и}$$
$$x \not\equiv \pm 1 \; mod \; p \qquad (5)$$

Такова число x е прието да се нарича свидетел за делимост.

Принципът на идея за модел на решение, позволяваща подобряване на резултатите на най-широко използваният алгоритъм в практиката за определяне на делимост на число, този на Милър-Рабин, е разработен и е предложен на научна конференция DIGILIENCE 2020 (Варна) и публикуван в международно издание „Information & Security: An International Journal“ [71].

Модела на решение разглежда възможността за преход между отделни мултипликативни подгрупи G_i, формирани от собствен генератор g_i и проверяваното число p. Идеята за тази реализация на предлагания модел на решение се основава на евристичен алгоритъм, при който е използвано линейно диофантово уравнение:

$$d_x \,.\, d_y - p.k = \; d_z \qquad (6)$$

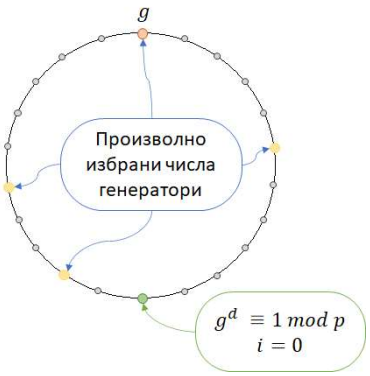
където $d_i = g^i \ mod\ p$ и всяко $d_i \in Z_p$ е елемент на пръстен с ред $\#O_{(g,p)}$, получен с генератор g. В частен случай, когато $d_x = {d_y}^{-1}\ mod\ p$ то стойността на d_z е 1.

Чрез евристичен подход е установено, че при $d_z = 1$ и ред на пръстен $\#O_{(g,p)} < p - 1$, можем да получим нова стойност на ${d_x}^*$, за която $\left(\frac{d_x}{p}\right) \neq \left(\frac{{d_x}^*}{p}\right)$. Тази нова стойност на ${d_x}^*$, използвана като генератор, много често формира пръстен с множество от елементи, които напълно или частично се различава от множеството елементи на пръстена формиран с генератор d_x. Това може да се реализира чрез следните итерации за намиране на подходяща стойност на ${d_x}^*$:

Input:	$\boldsymbol{d_x, p}$
Output:	$\boldsymbol{{d_x}^*}$
(1)	${d_x}^* \leftarrow d_x$
(2)	$m \leftarrow \left(\frac{d_x}{p}\right)$
(3)	**do**
(4)	**solve:** ${d_x}^* . d_y - p.k = d_z$
(5)	${d_x}^* \leftarrow k$
(6)	**while** $m \neq \left(\frac{{d_x}^*}{p}\right)$

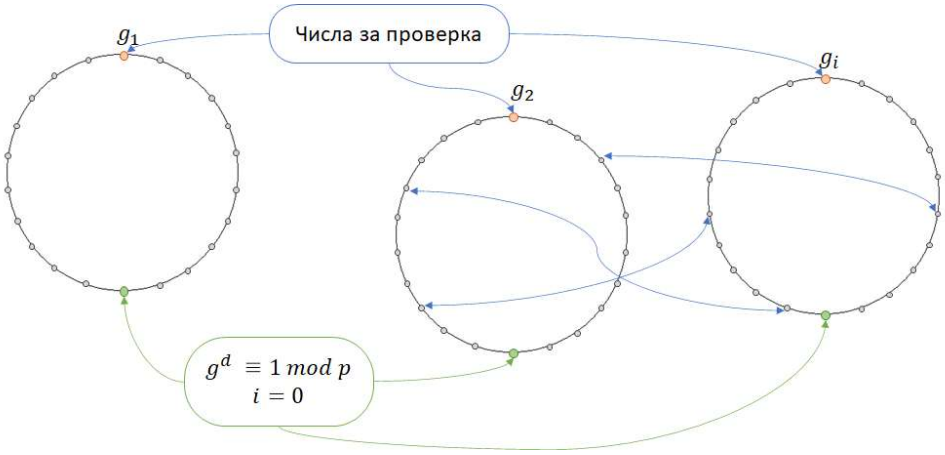
В практиката с цел бързо определяне делимост на число чрез използване на Милър-Рабин не се проверяват всички x в интервала $1 < x < 4.log^2 p$, а се избират произволни по стойност. При работа с големи числа вероятността да се избере число от същото множество, генерирано от конкретния генератор g е огромен, което е следствие от това, че се работи с много големи по стойност числа използвани за модул. Това създава възможност за пермутация на множеството и резултатът от проверката, чрез алгоритъма на Милър-Рабин, ще бъде същият без да допринася с нищо при проверката за делимост. Това графично е показано на фигура 3.

Ако се използва механизмът, при който, вместо случайни стойности на числа генератори (след първия генератор) се използват такива генератори, производни на първия или всеки следващ, но формиращи пръстени, чието множество от елементи изцяло не съвпада или има частично сечение с множеството елементи на предходните, могат да се проверят множества от числа, които дори да покрият напълно Z_p, за значително по-малко на брой итерации проверки по метода на Милър-Рабин. Този подход на итеративен модел е онагледен графично на фигура 4.



Фиг. 3

Изобразява резултата, избраните числа за проверка (при стандартен тест с Милър-Рабин) да са част от един и същи пръстен, което реално не повишава достоверността на теста



Фиг. 4

Изобразява идеята за използването на такива числа за проверка посредством теста на Милър-Рабин, които да са част от пръстени с различни по състав елементи

Стъпките на алгоритъм, реализиращ предлагания модел на решение (чиято идея графично е изобразена на фигура 4) са следните:

Input:	**number p**
Output:	**COMPOSIT, PRIME**
(1)	**pick up small prime number** q **such that** $a = p \, mod \, q > 1$
(2)	$b \leftarrow a^{p-1} \, mod \, p$
(3)	**if** $b > 1$f
(4)	**output: COMPOSIT**
(5)	$JacobiSymbol = 1$
(6)	$f \leftarrow (p-1)$
(7)	**for each** $i \in 0..4$
(7.1)	$b \leftarrow f \, / \, 2$
(7.2)	**if** $JacobiSymbol = 1$
(7.3)	$g \leftarrow$ ***GetPRU*** (a,p)
(7.4)	**if** $JacobiSymbol = -1$
(7.5)	$g \leftarrow$ ***getNotPRU*** (a,p)

(7.6)	$JacobiSymbol \leftarrow \left(\frac{g}{p}\right)$
(7.7)	**if** $JacobiSymbol = 0$
(7.8)	**output: COMPOSIT**
(7.9)	$LegendreSymbol \leftarrow g^b \bmod p$
(7.10)	**if** $JacobiSymbol <> LegendreSymbol$
(7.11)	**output: COMPOSIT**
(7.12)	**while** $(LegendreSymbol = 1)\ and\ (b \bmod 2 = 0)$
(7.12.1)	$b \leftarrow b\ /\ 2$
(7.12.2)	$LegendreSymbol \leftarrow g^b \bmod p$
(7.12.3)	**if** $(LegendreSymbol > 1)\ and\ (LegendreSymbol < f)$
(7.12.4)	**output: COMPOSIT**
(7.13)	$a \leftarrow [g * modfK(g,p)] \bmod p$
(7.14)	**If** $a = 1$
(7.15)	$a \leftarrow g$
(8)	**output: PRIME**

Псевдо кода на функциите, използвани в алгоритъма е следната:

	function $modK$ **(in** $generator$**, in** $modulonumber$**)**
(1)	$a \leftarrow generator^{-1} \bmod modulonumber$
(2)	**return** $\lfloor((a\ .generator) - 1)/modulonumber\rfloor$

	function $getNotPRU$ **(in** $generator$**, in** $modulonumber$**)**
(1)	$g \leftarrow generator$
(2)	$U \leftarrow modulonumber$
(3)	**if** $\left(\frac{g}{U}\right) = -1$
(4)	$g \leftarrow g^2 \bmod modulonumber$
(5)	$i \leftarrow 0$
(6)	$m \leftarrow g$
(7)	**do**
(8)	$i \leftarrow i + 1$
(9)	$m \leftarrow U - m$
(10)	$m \leftarrow modK(m,U)$
(11)	**if** $m < 2$
(12)	$i \leftarrow 0$
(13)	$m \leftarrow modK(g,U)$
(14)	$g \leftarrow (m+g)^g \bmod U$
(15)	**while** $\left(\frac{g}{U}\right) \in 0..1$
(16)	**return** m

	function $GetPRU$ **(in** $generator$**, in** $modulonumber$**)**
(1)	$g \leftarrow generator$
(2)	$U \leftarrow modulonumber$
(3)	**if** $\left(\frac{g}{U}\right) = -1$
(4)	$g \leftarrow g^2 \bmod modulonumber$
(5)	$e \leftarrow U \bmod 5$
(6)	**if** $e \in \{2..3\}$
(7)	**return** $5\ .g \bmod U$
(8)	$e \leftarrow U \bmod 6$

```
(9)   if e = 5
(10)      return −3 . g  mod U
(11)  e ← U mod 8
(12)  if e ∈ {3,5}
(13)      return 2 . g  mod U
(14)  if e = 7
(15)      return −2 . g  mod U
(16)  i ← 0
(17)  m ← g
(18)  do
(19)      i ← i + 1
(20)      m ←  U − m
(21)      m ← modK(m, U)
(22)      if m < 2
(23)          i ← 0
(24)          m ← modK(g, U)
(25)          g ← (m + g)^g mod U
(26)  while  (g/U) ∈ 0..1
(27)  return m
```

2.2 Модел на решение, позволяващ прилагане на алгоритъма на Силвър-Похлиг-Хелман без ограничаващи условия.

Всеки вид атака спрямо система за криптографска защита е конструирана така, че да си осигури необходимо количество информация и чрез допълнителен анализ и в повечето случай в комбинация със сериозен математически апарат да реализира извличане на явната информация. В много редки случаи това се постига без извличане на ключа за криптиране.

Алгоритмите с публичен ключ теоретично са по-лесни за атака, отколкото алгоритмите, опериращи със симетрични ключове, тъй като недоброжелателят лесно получава копие на публичния ключ, използван за криптиране на информацията. Част от усилията на недоброжелателя са улеснени, защото съобщението в голяма степен дава информация кой е използвания алгоритъм за криптиране и е предварително известна големината на ключа. Атаките на алгоритъма с публичен ключ се категоризират в две групи: атаки с ключово търсене и аналитични атаки [68].

Атаките с ключово търсене са по-популярния вид атаки за използване срещу криптографски системи с публичен ключ, защото те са най-лесно разбираеми. Тези атаки се реализират чрез извличане на частния ключ от свободно достъпния публичен ключ [42, 56].

Атаки срещу системата с публичен ключ, изградена на база RSA се извършват чрез опит да се разложи модулното число N на прости множители. Това модулно число е достъпно за атакуващият, защото то е домейн и на двете части на ключа (частна и публична). Веднъж разложено на множители модулното число N, позволява лесно да се изчисли частният ключ.

В тази част от дисертационния труд е предложен модел на решение на система конгруентни уравнения, който използван в алгоритми за възстановяване на частния ключ чрез ефективно разлагане на модулно число N позволява възстановяването на частния ключ d. Това е възможно за реализация при алгоритми

като този на Силвър-Похлиг-Хелман при който е необходимо решаването на система конгруентни уравнения.

В тази част на дисертационния труд е направено представяне на алгоритъма, предложен от Силвър-Похлиг-Хелман г.

Алгоритъм на Силвър-Похлиг-Хелман ...

В реализацията на алгоритъма на Похлиг-Хелман се използва следните класически алгоритми: изчисляване на най-голямо общо кратно, повдигане на степен в модулната аритметика ($z = x^n \ mod \ p$), Изчисляване на „инверсен модул“ ($z = x^{-1} \ mod \ p$) и решаване на система от конгруентни уравнения.

Решаване на система от конгруентни уравнения

За намиране на решение на системи от този вид, от векове се прилага алгоритъм известен като „Китайската теорема“ и за който се счита, че е създаден през 2ри век преди Новата Ера от Сун Дзъ.

Ограничението при прилагане на „Китайската теорема“ е, че модулните числа трябва да са взаимно прости, което налага пълно разлагане на реда на групата на прости множители. Това ограничение може успешно да бъде преодоляно, ако се приложи алгоритъм, който позволява решаване на система конгруентни уравнения без това ограничаващо условие. Алгоритъм, предлагащ възможност за решаване на системи конгруентни уравнения без наличие на ограничаващи условия бе разработен и предложен на научна конференция „Multimedia Communications, Services and Security. MCSS 2020“ [35].

Мотивацията за това изследване и разработка е свързана с две причини: 1) да е възможно решаване на системи от уравнения за конгруентност в случаите, когато модулните числа не са взаимно прости; 2) да е възможно прилагането на решение в системи за паралелни изчисления на големи числа.

За да бъде изяснена идеята на този подход, ще изведем математическо уравнение, което ще бъде основа на този алгоритъм.

Ако ни е дадена система конгруентни уравнения:

$$\begin{aligned} x &\equiv r_1 \ mod \ u_1 \\ x &\equiv r_2 \ mod \ u_2 \\ x &\equiv r_3 \ mod \ u_3 \\ &\cdot \\ &\cdot \\ &\cdot \\ x &\equiv r_i \ mod \ u_i \end{aligned} \qquad (7)$$

и изразим първите две уравнения като: $x_0 = r_1 + k_1.u_1$ и $x_0 = r_2 + k_2 \, .u_2$ ще имаме решение най-малко по стойност x_0. Това е постижимо, ако знаем стойностите на k_1 и k_2, където $1 \leq k_1 < u_2$ и $1 \leq k_2 < u_1$. Стойността на x_0 ще е решение на първите две уравнения от системата.

Ако умножим двете страни на уравненията ще получим:

$$x_0{}^2 = (r_1 + k_1.u_1)(r_2 + k_2.u_2) \qquad (8)$$

$$x_0{}^2 = \ u_1.u_2.k_1.k_2 + k_1.u_1.r_2 + k_2.u_2.r_1 + r_1.r_2 \qquad (9)$$

Ако изчислим по модул A двете страни на (9), където $A = u_1.u_2$, ще получим следното тъждество:

$$x_0{}^2 \equiv (u_1.u_2.k_1.k_2 + k_1.u_1.r_2 + k_2.u_2.r_1 + r_1.r_2)\ mod\ A \quad (10)$$

В това уравнение стойността на $u_1.u_2.k_1.k_2\ mod\ A$ е равна на 0, следователно:

$$x_0{}^2 \equiv (k_1.u_1.r_2 + k_2.u_2.r_1 + r_1.r_2)\ mod\ A \quad (11)$$

По дефиниция на задачата $r_1 + k_1.u_1 = r_2 + k_2\ .u_2$, от където можем да изразим $k_1 = \frac{k_2.u_2 + r_2 - r_1}{u_1}$ и да го заместим в уравнение (11), от което следва:

$$x_0{}^2 \equiv \left(\frac{k_2.u_2 + r_2 - r_1}{u_1}\ .u_1.r_2 + k_2.u_2.r_1 + r_1.r_2\right)\ mod\ A \quad (12)$$

$$x_0{}^2 \equiv (k_2.u_2.r_2 + r_2{}^2 - r_1.r_2 + k_2.u_2.r_1 + r_1.r_2)\ mod\ A \quad (13)$$

$$x^2 \equiv [k_2.u_2.(r_2 + r_1) + r_2{}^2]\ mod\ A \quad (14)$$

От тъждество (14) ние можем да изведем:

$$x^2 \equiv [k_2.u_2.(r_2 + r_1) + r_2{}^2]\ mod\ u_1 \quad (15)$$

Тъй като $x_0 \equiv r_1\ mod\ u_1$, следователно $x_0{}^2 \equiv r_1{}^2\ mod\ u_1$ заместваме $z = x^2 = r_1{}^2\ mod\ u_1$ в (15) и получаваме:

$$z \equiv [k_2.u_2.(r_2 + r_1) + r_2{}^2]\ mod\ u_1 \quad (16)$$

Следователно, за да изчислим k_2 трябва да се реши следното диофантово уравнение:

$$b.k_2 - u_1.y = z - r_2{}^2 \quad (17)$$

където $b = u_2.(r_2 + r_1)$.

Използваме най-малкото по стойност положително решение за k_2 и го полагаме в $x_0 = r_2 + k_2\ .u_2$, за да получим решение за x_0, което е решение на първите две уравнения. Уравнение (17) винаги има решение, защото $z - r_2{}^2$ винаги се дели на $GCD(b, u_1)$. Доказателство за това е изведеното конгруентно уравнение (16).

Използвайки уравнение (17) в следващия алгоритъм, ние можем да решим цялата система конгруентни уравнения (7). Алгоритъмът е приложим за всички случаи без значение дали модулните числа u_i, u_j $(i \neq j)$ са взаимнопрости. Вследствие на това не се налага да се разлагат u_i на прости множители. Това позволява ускоряване на процеса за решаване на системи конгруентни уравнения.

Стъпки на алгоритъма, използващ представения математически апарат:

Input:	$Reminders\ [r_0, r_1, r_2, \dots, r_n]$ $modNumbers\ [u_0, u_1, u_2, \dots, u_n]$
Output:	x_0 - **минималната стойност на** x
(1)	$uniqLCM \leftarrow u_0$
(2)	$bValue \leftarrow r_0$
(3)	**for each** $i \in 1..n$
(4)	$m_1 \leftarrow uniqLCM$
(5)	$uniqLCM = (uniqLCM\ .\ u_i)$
(6)	$z \leftarrow (bValue^2\ mod\ m_1) - r_i{}^2$
(7)	$b \leftarrow (bValue + r_i)\ .u_i$

(8)	**solve:** $b \,.\, k_2 - m_1 \,.\, y = z$
(9)	$bValue \leftarrow\ m_2 \,.\, minpositive(\, k_2) + a_2$
(10)	**output:** $x_0 \leftarrow bValue$

Повторенията на цикъла на предложения алгоритъм са равни на броя на уравненията в системата минус едно. В този предложен алгоритъм стойностите на числата се увеличават с всяка итерация. В класическия алгоритъм всички математически операции трябва да се извършват с двоичен размер на числата, близък до произведението на всички модулни числа m_i. Единственият недостатък на описания по-горе алгоритъм е, че ако някоя двойка u_i и u_j $(i \neq j)$ не е взаимно проста, тогава x няма да бъде минималната стойност решение на системата. За преодоляване на този недостатък бе разработен следващия алгоритъм.

Input:	$Reminders\ [r_0, r_1, r_2, \dots\ , r_n]$ $modNumbers\ [u_0, u_1, u_2, \dots\ , u_n]$
Output:	x_0 - **минимална стойност на търсеното** x
(1)	$uniqLCM \leftarrow\ u_0$
(2)	$bValue \leftarrow\ r_0$
(3)	**for each** $\ i\ \in 1..n$
(4)	$m_1 \leftarrow uniqLCM$
(5)	$g \leftarrow GCD(uniqLCM, u_i)$
(6)	$uniqLCM \leftarrow (uniqLCM \,.\, u_i)\ div\ g$
(7)	**if** $(u_i\ > m_1)$
(8)	$m_2 \leftarrow m_1$
(9)	$a_2 \leftarrow bValue$
(10)	$m_1 \leftarrow u_i$
(11)	$a_1 \leftarrow r_i$
(12)	**else**
(13)	$a_1 \leftarrow bValue$
(14)	$m_2 \leftarrow u_i$
(15)	$a_2 \leftarrow r_i$
(16)	$z \leftarrow\ ({a_1}^2\ mod\ m_1) - {a_2}^2$
(17)	$b \leftarrow [(a_1 + a_2) \,.\, m_2]$
(18)	**solve:** $b \,.\, k_2 - m_1 \,.\, y = z$
(19)	$bValue \leftarrow\ m_2 \,.\, minpositive(\, k_2) + a_2$
(20)	**if** $(bValue\ mod\ m_1\ <>\ a_1)$
(21)	$t = a_1 - bValue$
(22)	$w = m_2 \,.\, minpositive(\, k_2)$
(23)	**solve:** $w \,.\, n - m_1 \,.\, y = t$

(24)	$bValue = bValue + w.minpositive(n)$
(25)	**output:** $x_0 \leftarrow bValue$

За да бъде демонстрирано функционирането на алгоритъма е даден пример в „Приложение 2“. В този пример модулните числа не са взаимно прости две по две.

Предлаганият модел на решение е приложим при реализацията на алгоритъма на Силвър-Похлиг-Хелман без ограничаващи условия, които трябва да са спазени, ако се прилага „Китайската теорема“ за решаване на система от конгруентни уравнения.

2.3 Изводи.

- Модел на решение, позволяващ подобряване на оценката за делимост на число при използване на алгоритъма на Милър Рабин в процеса на генериране на прости числа за криптографски ключове води до повишаване на устойчивостта на функциониране на RSA базирани криптографски системи.
- Модел на решение, позволяващ прилагане на алгоритъма на Силвър Похлиг-Хелман без ограничаващи условия може да бъде използван за намаляване броя на итерациите необходими за постигане на по-бърза и ефективна атака спрямо RSA базирани криптографски системи и създава предпоставки за понижаване устойчивостта на функционирането на системи, базирани на RSA.
- Атаки върху един единствен ключ не позволява обхватност на атаката. Атаки, използващи аналитичен подход са широко обхватни и позволяват компрометиране на конкретен криптографски алгоритъм или на цяла група алгоритми.

ТРЕТА ГЛАВА

МЕТОДИКА ЗА ИЗПОЛЗВАНЕ НА КЛЕПТОГРАФИЯ В RSA БАЗИРАНА КРИПТОГРАФСКА СИСТЕМА.

Съгласно правилата на публичната криптография, необходимите условия за постигане на достатъчна сигурност са спазване на стандартите при генериране на ключове и надеждното и сигурно съхраняване на личните части на всеки от тях. Изпълнението на всичко това може да не е гарант за сигурността на RSA базирана криптографска система. Сигурността може да бъде компрометирана лесно, при условие че създателя на средството, с което се генерират ключове, разполага с информация, позволяваща му разлагане на модулното число N с бързина приблизително равна на работа в реално време, чрез прилагане на клептография [38].

В тази глава са изложени математически основи и модел на решение на нова идея на клептографски алгоритъм за атака. Направена е оценка за възможността за приложимост и сравнителен анализ със съществуващи алгоритми. Анализирана е степента на устойчивост от разкриване и са

определени техническите параметри, които характеризират предложеният алгоритъм.

3.1 Математически основи и модел.

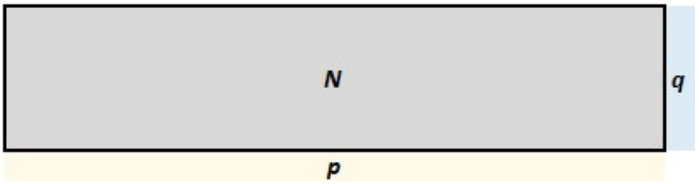
Идеята за използване на точни квадрати, имащи връзка към дадено съставно число е създадена и предложена за първи път от Пиер дьо Ферма през 1643 година [22]. От тогава е известно, че точните квадрати имат ефективно приложение в процеса на разлагане на съставни числа на множители. Използвайки това познание ще представим с формули и графично, някой нови гледни точки, свързани със съставните числа и връзката на точните квадрати с тях.

Нека $N = p.q = t^2 + r$ или $N \bmod t = r$, където $t = \lfloor \sqrt{N} \rfloor$ и $N + R = (t+1)^2$. От това представяне можем да изведем следните уравнения:

$$(t+1)^2 = t^2 + 2t + 1 = N + R \quad (18)$$

$$2t - r = R - 1 \quad (19)$$

Представянето на числото N и изведените до момента формули можем да визуализираме графично на следващите фигури, където числото N е изобразено със сив цвят и е представено като правоъгълник, чието лице е произведение на двете му страни: число p (в светло жълт цвят) и число q (в бледо син цвят). Това графично представяне напълно съответства на представянето на съставно число $N = p.q$.



Фиг. 6

Графично представяне на числото N като произведение на p и q

Чрез фигура 7 са онагледени графично връзката на числата t (в бледо зелено) формиращо квадрат със страна $t+1$ и числата R (бледо оранжево) и r (в тъмно синьо).

Ще направим второ представяне на съставното число N, $p = t + x$ и $q = t - y$, където $p > q \rightarrow x > y$ (фигура 7), от което следва, че:

$$N = (t+x)(t-y) \quad (20)$$

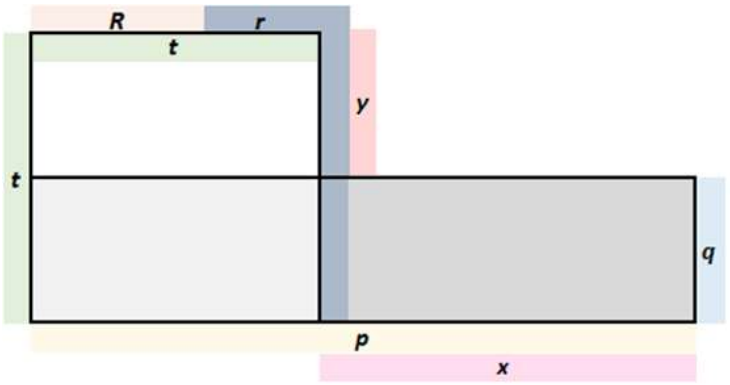
където : $p = (t+x)$, $q = (t-y)$

$$N = t^2 + tx - ty - xy \quad (21)$$

$$N - t^2 = t(x-y) - xy \quad (22)$$

$$t(x-y) - xy = r \quad (23)$$

$$t = \frac{r + xy}{(x-y)} \quad (24)$$



Фиг. 7

Графично представяне на математическата зависимост на числата N, p, q ***и*** t, r, R, x, y

От уравненията представени до тук можем да изведем следващите формули:

$$p + q = 2t + x - y \quad (25)$$

$$p - q = x + y \quad (26)$$

Ако образуваме квадрат със страна $\frac{p+q}{2}$ и от лицето му извадим лицето на квадрат с дължина на страната t, ще получим разлика, която ще обозначим с v : $\left(\frac{p+q}{2}\right)^2 - t^2 = v$, следователно $\left(\frac{p+q}{2}\right)^2 = t^2 + v$.

По дефиниция, ако N е модулно число на ключ за RSA, то това число е съставно, получено с умножение на две прости числа p и q, които по стойност са големи числа, много по-големи от 2. След като са прости числа и по-големи от 2, то те са нечетни по стойност, следователно сборът им се дели на 2 без остатък и $\frac{p+q}{2}$ е цяло число.

Лицето на квадрата със страна $\frac{p+q}{2}$ е по-голямо от това на квадрата със страна t, защото по дефиниция $x > y$ (виж уравнение 20 и по условие $p > q$). След като $\frac{p+q}{2}$ е по-голямо по стойност от t можем да запишем равенството:

$$\frac{p+q}{2} = t + m \quad (27)$$

На база уравнения (25) и (27) е в сила равенството:

$$\frac{2t + x - y}{2} = t + m \quad (28)$$

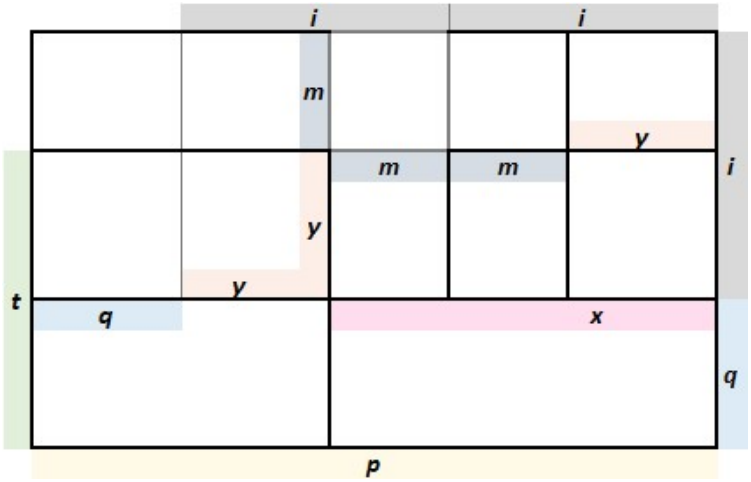
Опростявайки (28) получаваме:

$$x - y = 2m \quad (29)$$

За по добра представа ще използваме следващите две фигури (8 и 9), които показват нагледно връзките между формулите. На тези графики числото m използвано в уравнения (27, 28, 29) е изобразено с тъмно синьо-сив цвят, числото y с бледо оранжев цвят, а числото x с бледо розов цвят.

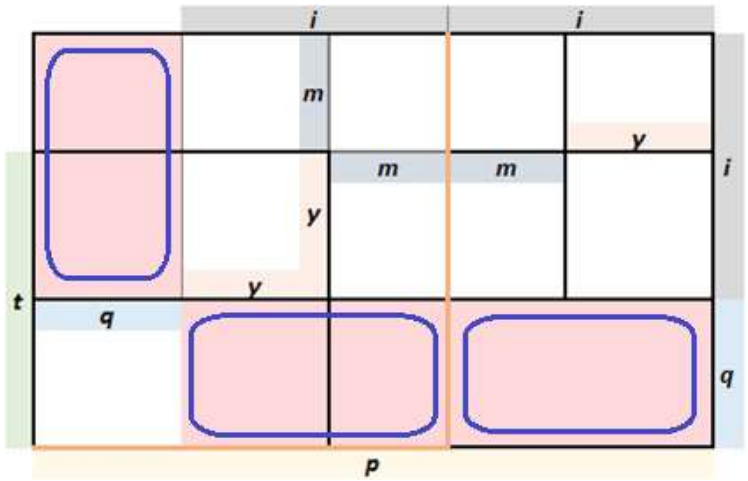
Ако означим с $i = y + m$ и разгледаме фигура 9, където със син цвят са отбелязани 3 области с равни лица (3 правоъгълника оцветени с светло червен цвят и всеки със страни q и $i = y + m$), можем да изведем уравнение (30), което е производно на всички описани до момента:

$$(t+m)^2 - i^2 = N \tag{30}$$



Фиг. 8

Графично представяне на математическата зависимост на числата N, t, x, y и m, i



Фиг. 9

Графично представяне на математическата зависимост между $(t+m)^2$, i^2 и числото N

Ако представим $m = g + 1$ и заместим в уравнение (29) ще получим:

$$g = \frac{x - y - 2}{2} \tag{31}$$

Замествайки m в (30) ще получим:

$$(t + g + 1)^2 - i^2 = N \tag{32}$$

Ако изразим i^2 чрез $i^2 = (t + g + 1)^2 - N$ и опростим ще получим уравнение, която изразява i чрез g:

$$i^2 = (t+1)^2 + 2\,.\,g\,.(t+1) + g^2 - N \tag{33}$$

От (18) знаем, че $N + R = (t+1)^2$, замествайки $(t+1)^2 - N$ в (33) получаваме:

$$i^2 = g^2 + 2\,.\,g\,.(t+1) + R \tag{34}$$

Изводите, които можем да си направим от (32) и (34) са два: 1) при твърдо зададени стойности на t и R, стойността на N зависи от величината на g; 2) минимална стойност на g имаме, тогава когато разликата между p и q е 2, тогава $g = 0$.

По дефиниция, ако разполагаме с число N, лесно изчисляваме t и R, защото от формулите знаем, че стойността на $t = \lfloor \sqrt{N} \rfloor$ или $N \bmod t = r$ и $R = (t^2 + 1) - N$. От казаното следва че, ако t и R са известни величини, то разлагането на съставно число N зависи от намиране на цяло число g, което да удовлетворява едновременно уравнения (32) и (34). От изведения до тук математически апарат е вярно твърдението, че стойността на g е право пропорционална на разликата между p и q.

В подхода, предложен от Ферма се търсят две числа разликата от квадратите, на които да е равна на числото N. В изведените до тук уравнения тези две числа са еквивалентни на i и сбора $t + g + 1$ (виж уравнение 32). От така използвания подход и изведените уравнения е видно, че тези две числа са в пряка зависимост от стойността на число g (виж уравнения 32 и 34) и са значително по-големи по стойност, което означава, че търсене на стойност на числото g е значително по-бърз процес и би довел до разлагане на N за много по-малко на брой итерации. Броят на числата, които могат да удовлетворят уравнения (32) и (34) зависи от броят двойки множители, чието произведение е N. В случай че N е произведение на два прости множителя, то стойността на g удовлетворяваща тези уравнения е само една. От изведеният математически апарат може да се направи и още едно вярно заключение: при числа N получени при умножението на повече от два прости множителя, най-малко по стойност g_i ще имаме тогава когато разликата на двойката $p_i - q_i$ е най-малка от всички възможни двойки за p_i и q_i, които като произведение са равни на N.

На база създаденият математически апарат може да се потърси отговор на въпроса: „съществува ли начин, чрез който стойността на g която да удовлетвори двете уравнения, да бъде намалена“. Отговора на този въпрос е **„ДА“**. Това може да се постигне чрез умножаване на N с $n = a.b$ и ако $\frac{a}{b} \approx \frac{p}{q}$, то за $M = N.n = p.q.a.b$ ще имаме $g_M < g_N$. Дори и при не напълно съвпадение на стойностите на отношенията ($\left\lfloor \frac{a}{b} \right\rfloor = \left\lfloor \frac{p}{q} \right\rfloor$), стойността на g_M може да е 0. С други думи ще имаме ново по-голямо съставно число M с множители $a.p \approx b.q$, за което $\left\lfloor \frac{a.p}{b.q} \right\rfloor \approx c$, където $c \ll q < p$ и $g_M < g_N$.

Разполагайки с g_M можем да разложим на прости множители M и така можем да изчислим стойностите на p и q чрез намиране на най-голям общ делител $GCD(f, N)$ където f е множител на M. Вариантите за този множител могат да бъдат следните комбинации: $f = p.a$, $f = p.b$, $f = q.a$ или $f = q.b$.

От това следва твърдението, че ако множителите p и q на N генерират малко по стойност g, е много вероятно чрез пълно изчерпване за g, числото N да бъде разложено.

От всичко изложено до тук може да се твърди, че за сигурността на криптографски системи, базирани на RSA, не е достатъчно покриването на условията на стандарт „SP 800-56B“ от 2020г. на NIST и стандарт „TS 102 176-1 V2.0.0“ на ETSI от 2007 година. Според тези стандарти да припомним:

- $p - q > 2^{nBits-100}$ по NIST
- $0.1 < |log_2 p - log_2 q| < 30$ по ETSI

Дори и да са изпълнени, тези условия, ако с малки стойности на a и b получим $M = n.N$, за което $g_M \ll q < p$, можем с пълното изчерпване за n, комбинирано с пълното изчерпване на g_M да постигнем бързо разлагане на N, ако $g_M \,.\, n \ll q < p$. Изказано с други думи, „отношението“ на p и q е изключително важно за сигурността на системи, базирани на RSA. Процеса на откриване на баланс между n и двата множителя на N, е наречен: **„Синтез на числов баланс“ („Fusion of Number Balance“ - FNB)**.

Значимостта на отношението между двата множителя p и q на модулно число N участващо във формирането на RSA ключ и практическото използване на алгоритъм „Синтез на числов баланс“ е демонстрирано в пример, предоставен в „Приложение 3“.

В заключение на тази точка може да се изкаже твърдението, че „Синтез на числов баланс“ не представлява универсален алгоритъм за разлагане на числа на прости множители. Може да бъде използван с пълно изчерпване на стойности за n само за съставни числа, при които отношението на множителите p и q го позволява. Той обаче е нагледен пример за това, че условията за проверка на произволно генерирани прости числа, за елементи на ключове за RSA базирана криптографска системи, трябва да бъдат преразгледани и да се включи допълнително условие проверяващо големината на g.

3.2 Практическа реализация на клептографски алгоритъм за RSA базирана криптографска система.

С цел изясняване на степента на потенциален риск за устойчивостта на криптографски системи с публичен ключ (базирани на RSA), посредством въздействие върху криптографските примитиви, чрез описаните в предходната точка математически основи, е разработен вариант на клептографски алгоритъм. На база изследване на телеметричните му параметри, потенциалните възможности за скрито функциониране и др., са направени оценки за нивата на уязвимост на сигурността, които могат да бъдат реализирани при евентуално негово използване.

За да отсъстват предпоставки за наличие на слабости при генерирането на прости числа, е създаден алгоритъм за тяхното генериране, който като резултат създава произволно число, което се проверя за делимост. Входен параметър на алгоритъма е дължината в битове, която числото трябва да има. Псевдо кода на алгоритъма има следния вид:

Input:	l **– необходима дължина в битове**
Output:	p **– просто число**
(1)	$t \leftarrow 2^l$
(2)	$z \leftarrow 0$
(3)	**do**
(4)	$z \leftarrow z\ shl\ 8 + randomByte$
(5)	**while** $z > t$

(6)	$b \leftarrow z\ shr\ (bitlengt(z) - l)$
(7)	**if** $b\ mod\ 2 = 0$
(8)	$b \leftarrow b + 1$
(9)	**while** $b\ not\ prime$
(10)	$b \leftarrow b + 2$
(11)	**output:** $p \leftarrow b$

Алгоритъма не се базира на принципи и методи за бързо генериране на прости числа, защото при повечето от тях в процеса на генериране се използва крайно множество от по-малки по стойност прости числа. Това ограничава множеството, в което попада крайното генерирано число. Предложеният алгоритъм е итеративен и броят на итерациите, които ще са необходими до откриване на просто число ще са равни на $\frac{(p-b)}{2}$ (съгласно означението на променливите във псевдо кода). Така може да се твърди, че сложността на изпълнение на този алгоритъм е произведение на $\frac{(p-b)}{2}$ и сложността на алгоритъма, който се използва за проверка на делимост (сложност на Miller-Rabin primality test например).

С цел по-голяма яснота и удобство, идеята за клептографски алгоритъм ще бъде описана схематично и алгоритмично, което да улеснява последващите оценки и анализи. За краткост предлагания алгоритъм е наречен „**gBaseKleptoRSA**“. Означенията на числата в описанията ще съответстват по смисъл на означенията в предходната точка**.**

За реализация на атака с gBaseKleptoRSA нападателя притежава таен ключ n. Този ключ е със стойност $n = a.b$, където a и b са взаимно прости числа ($GCD(a\ ,\ b) = 1$) и с големина по голяма от 2^{32}. Размера на a и b е приет да бъде $> 2^{32}$, защото е логично да използваме такива двойки $[a, b]$, така че $n = a\ .b$ да е с достатъчна размерност в битове, която да е сравнително голяма, за да се реализира пълно изчерпване за нейното разкриване. Числата a и b ще послужат за формиране на твърдо отношение между генерираните множители на N.

Търсената дължина на ключа ще означим с k. Бинарната дължина на n ще означим с l. За да получим стойностите на множителите за всеки клептографски ключ, който генерираме ще използваме случайно генерирано просто число u с бинарна размерност $\frac{k-l}{2}$. За всеки ключ стойността на u е отделно генерирана, не се повтаря и не се съхранява. Получаваме множителите, чрез следните пресмятания:

$$p = NextPrime(u.a) = Prime(u.a + r_p)$$
$$q = NextPrime(u.b) = Prime(u.b + r_q)$$

Това означава, че стойността на p е следващото просто число по-голямо от $u.a$, а за q е следващото просто число по-голямо от $u.b$. Максималната приблизителна големина на стойностите на r_p и r_q можем да изчислим използвайки формулата, с която се пресмята приблизителния брой на простите числа по-малки от x: ${}^{x}/_{\ln x}$.

Въпреки, че тази формула не дава точен резултат, а приблизителен и години изследване на проблема за определяне бройката на простите числа до стойност

на число x не са дали резултат за изчисление с абсолютна точност, за целите които ни е необходима ние можем да я използваме. С нейна помощ получаваме: $r_p \approx \frac{p}{\ln p} - \frac{u.a}{\ln u.a}$ и $r_q \approx \frac{q}{\ln q} - \frac{u.b}{\ln u.b}$. Стойностите на r_p и r_q влияят върху скоростта на генериране на p и q. Приблизителният брой на необходимите проверки за делимост на p и q ще бъде $r_p \,.\, r_q$ = брой тестове за делимост.

По този начин, чрез тайната стойност на числото n ще получим стойност на $q_M = 0$ за числото $M = n.N$. Така за атакуващия остава да извърши следните пресмятания:

$$M = n.N$$
$$t_M = \lfloor \sqrt{M} \rfloor$$
$$i_M = \sqrt{(t_M + 1)^2 - M}$$
$$f = t_M - i_M + 1$$
$$p = GCD(f, N) \quad q = N \; div \; p$$

Използваната публична експонента на генерирания ключ е входна величина, защото по стандарт тя трябва да е малко по стойност просто число с възможно най-малко на брой битове равни на 1. В практиката най-използваната стойност за публична експонента на RSA ключ, е:

$$e = 65537_{dec} = 100001_{bin}.$$

Псевдо кода на предлагания алгоритъм е :

Input:	$\boldsymbol{k}$ **– необходима дължина на** $\boldsymbol{N}$ **в битове** $\boldsymbol{e}$ **– публична експонента** $(\boldsymbol{a}, \boldsymbol{b}: \boldsymbol{n} = \boldsymbol{a}.\boldsymbol{b})$ **– таен ключ на атакуващия**
Output:	$\boldsymbol{p}, \boldsymbol{q}$ **– просто число** $\boldsymbol{N}$ **– модулно число** $\boldsymbol{d}$ **– личен ключ**
(1)	$n \leftarrow a \,.\, b$
(2)	$l \leftarrow BitSize(n)$
(3)	$u \leftarrow RandomPrime\left(\frac{k-l}{2}\right)$
(4)	$p \leftarrow NextPrime(u.a)$
(5)	$q \leftarrow NextPrime(u.b)$
(6)	**if** $(p-1)\,.\,(q-1) \; mod \; e = 0$
(7)	**goto: (3)**
(8)	**if** $p < q$
(9)	$z \leftarrow p$
(10)	$p \leftarrow q$
(11)	$q \leftarrow z$
(12)	$N \leftarrow p \,.\, q$
(13)	$d \leftarrow e^{-1} \; mod \; \varphi(N)$
(14)	**output:** p, q, N, d

За да бъде напълно и правилно функциониращ предложеният алгоритъм, в него е включена проверка за делимост, при която се тества дали стойностите на $\varphi(p)$ и $\varphi(q)$ са делими на публичната експонента e. Това е задължително условие,

защото ако не са взаимно прости то е невъзможно да бъде изчислен частният ключ, така диофантовото уравнение:

$$e\,.\,d - \varphi(N).\,t = 1$$

няма да има решение, защото $GCD(\varphi(N), e) > 1$.

На база описания математически апарат ще използваме псевдо код, за да опишем алгоритъма, чрез който атакуващия ще разложи модулното число N:

Input:	$\boldsymbol{N}$ **– модулно число** $\boldsymbol{e}$ **– публична експонента** $(\boldsymbol{a}, \boldsymbol{b}: \boldsymbol{n} = \boldsymbol{a}.\boldsymbol{b})$ **– таен ключ на атакуващия**
Output:	$\boldsymbol{p}, \boldsymbol{q}$ **– просто число** $\boldsymbol{d}$ **– личен ключ**
(1)	$M \leftarrow n.N$
(2)	$t \leftarrow \lfloor \sqrt{M} \rfloor$
(3)	$i \leftarrow \sqrt{(t+1)^2 - M}$
(4)	$p \leftarrow GCD(t - i + 1, N)$
(5)	$q \leftarrow {}^{N}/_{p}$
(6)	$d \leftarrow e^{-1}\, mod\, \varphi(N)$
(7)	**output:** p, q, d

Предложеният алгоритъм бе реализиран в програмен код и чрез създаденото софтуерно приложение бяха генерирани над 2000 броя ключове с размер 2048 бита. На база логически анализ и оценка на резултатите бе установено, че в не малка част от генерираните ключове 74.8%, стойността на най-старшия бит е със стойност 0. В тези случаи реалната големина на ключа не съответства на изискването на стандартите и той не е 2048 битов. Тази слабост на алгоритъма бе причина да се създаде нов алгоритъм с различен принцип на използване на генерирани прости числа, но със запазена математическа и алгоритмична основа на клептографски метод за атака. Тази версия на алгоритъма е с условно наименование **„gBaseKleptoRSA2“**.

В новия метод не се разчита само на една двойка a и b множители за една стойност на n, а се генерира множество от двойки a_i и b_i. Така процеса се разделя на два етапа. В първия се генерира базова таблица, която е с постоянни стойности на елементите. Това позволява тя да е част от хардуерен програмен код или предефинирани константи в софтуерна програма. Тя представлява множество от варианти за ключове на нападателя и позволява да бъде използвана за бързо разлагане на модулното число N на всеки генериран чрез „gBaseKleptoRSA2“ ключ. Описанието на семантиката на двата етапа е следното:

- В първи етап по зададени входни параметри $[a_0, b_0]$ се генерира множество $S = \{[a_i, b_i],\ i \in 0..80\}$, за генерирането на което, се използва твърдо зададена матрица $R_{x,y} = 9 \times 9$ елемента $[l, h]$:

$$\begin{bmatrix} R_{0,0}=[0,0] & \cdots [0,4][0,-1]\cdots & R_{0,8}=[0,-4] \\ \vdots & \ddots & \vdots \\ R_{8,0}=[-4,0] & \cdots [-4,4][-4,-1]\cdots & R_{8,8}=[-4,-4] \end{bmatrix}$$

Тя указва закона по който, чрез бинарни операции, преместване на ляво (shl) и преместване на дясно (shr), от $[a_0,b_0]$ ще се генерират стойностите на останалите елементи на множество S, използвайки следните операции:

$$x = i\ div\ 9$$
$$y = i\ mod\ 9$$
$$a_i = a_0\ shl\ R_{x,y\rightarrow}\ l\ \ if\ R_{x,y\rightarrow}l \geq 0$$
$$a_i = a_0\ shr\ R_{x,y\rightarrow}\ l\ \ if\ R_{x,y\rightarrow}l < 0$$
$$b_i = a_0\ shl\ R_{x,y\rightarrow}\ h\ \ if\ R_{x,y\rightarrow}h \geq 0$$
$$b_i = a_0\ shr\ R_{x,y\rightarrow}\ h\ \ if\ R_{x,y\rightarrow}h < 0$$

Така се създава възможността да се разполага с поне един елемент от S, чрез който ако имаме следното изчисление за модулно число: $N = a_i\ .\ b_i\ .u^2 = (a_i\ .\ u)\ .(b_i\ .\ u) = p\ .\ q$, да получим такова, за което най-старшият бит на зададената дължина на ключ да е равен на 1ца, с което да е изпълнено условието на стандартите.

- Във втори етап се генерира просто число u за конкретен RSA ключ. Това u е случайно генерирано, използва еднократно и не се съхранява. Стартираме търсене и изчисляваме:

$$p_i = u.a_i$$
$$q_i = u.b_i$$
$$N = p_i\ .\ q_i$$

Първи индекс i, за който получим 1ца за най-старши бит на N прекъсва търсенето и правим следните пресмятания:

$$D_a = random(1..97)\ shl\ d_a$$
$$p = NextPrime\big(u.(a_i + D_a)\big)$$
$$D_b = random(1..97)\ shl\ d_b$$
$$q = NextPrime\big(u.(b_i + D_b)\big)$$
$$N = p\ .\ q$$
$$d = e^{-1}\ mod\ \varphi(N)$$

Описанието на псевдо кода на алгоритмите за двата етапа на генериране на ключа са следните.

Алгоритъмът, реализиращ стъпките на първи етап за генериране за таблица с двойки $[a_i, b_i]$ е описан със следния псевдо код:

Input:	$\boldsymbol{R_{x,y}}$ **– матрица** $\boldsymbol{[a_0,b_0]}$ **– таен ключ на атакуващия**
Output:	$\boldsymbol{S = \{[a_i, b_i],\quad i \in 0..80\}}$
(1)	$S[0].a \leftarrow a_0$,
(2)	$S[0].b \leftarrow b_0$,
(3)	**for each** $i \in 1..80$
(4)	$x \leftarrow i\ div\ 9$

(5)	$y \leftarrow i \; mod \; 9$
(6)	**if** $R[x,y].l \geq 0$
(7)	$S[i].a \leftarrow S[0].a \;\; shl \; R[x,y].l$
(8)	**else**
(9)	$S[i].a \leftarrow S[0].a \;\; shr \; R[x,y].l$
(10)	**if** $R[x,y].h \geq 0$
(11)	$S[i].b \leftarrow S[0].b \;\; shl \; R[x,y].h$
(12)	**else**
(13)	$S[i].b \leftarrow S[0].b \;\; shr \; R[x,y].h$
(14)	**output:** S

За генериране на ключ се изпълняват стъпките, описани в псевдо кода на следващия алгоритъм:

Input:	$\boldsymbol{k}$ **– необходима дължина на** $\boldsymbol{N}$ **в битове** $\boldsymbol{e}$ **– публична експонента** $\boldsymbol{S} = \{[\boldsymbol{a_i}, \boldsymbol{b_i}], \; \boldsymbol{i} \in \boldsymbol{0}..\boldsymbol{80}\}; d_a, d_b \in 1..24$
Output:	$\boldsymbol{p}, \boldsymbol{q}$ **– просто число** $\boldsymbol{N}$ **– модулно число** $\boldsymbol{d}$ **– личен ключ**
(1)	$l \leftarrow BitSize(S[0].b)$
(2)	$u \leftarrow RandomPrime\big((k/2) - l\big)$
(3)	$m \leftarrow random(1..80)$
(4)	**for each** $i \in 1..80$
(5)	$f \leftarrow (m + i) \; mod \; 80$
(6)	$d \leftarrow u^2 \; . \; S[i].a \; . S[i].b$
(7)	**if** $bitlength(d) = k$
(8)	$D_a \leftarrow random(1..97) \; shl \; d_a$
(9)	$p \leftarrow NextPrime\big(u.(S[i].a + D_a)\big)$
(10)	$D_b \leftarrow random(1..97) \; shl \;\; d_b$
(11)	$q \leftarrow NextPrime\big(u.(S[i].b + D_b)\big)$
(12)	**if** $p < q$
(13)	$z \leftarrow p$
(14)	$p \leftarrow q$
(15)	$q \leftarrow z$
(16)	$N \leftarrow \; p \; . \; q$
(17)	$d \leftarrow \; e^{-1} \; mod \; \varphi(N)$
(18)	**output:** p, q, N, d

В таблична форма в „Приложение 4“ са дадени примери, резултат от изпълнението на алгоритъма за генериране на ключове, чрез прилагането на gBaseKleptoRSA2.

Алгоритъмът, който прилага в изпълнение атакуващия, за да възстанови частен ключ генериран с gBaseKleptoRSA2 е със следните итерации:

Input:	$\boldsymbol{N}$ **– модулно число на ключа** $\boldsymbol{e}$ **– публична експонента** $\boldsymbol{S} = \{[\boldsymbol{a_i}, \boldsymbol{b_i}],\ \boldsymbol{i} \in \boldsymbol{0..80}\}$; $d_a, d_a \in 1..24$
Output:	$\boldsymbol{p}, \boldsymbol{q}$ **– просто число** $\boldsymbol{d}$ **– личен ключ**
(1)	**for each** $i \in 0..80$
(2)	**for each** $wL \in 1..97$
(3)	$aV \leftarrow S[i].a + wL\ shl\ d_a$
(4)	**for each** $wH \in 1..97$
(5)	$bV \leftarrow S[i].b + wH\ shl\ d_b$
(6)	$n \leftarrow aV\ .\ bV$
(7)	$M \leftarrow n.N$
(8)	$t \leftarrow \lfloor \sqrt{M} \rfloor$
(9)	$i \leftarrow \left\lfloor \sqrt{(t+1)^2 - M} \right\rfloor$
(10)	**if** $i^2 = (t+1)^2 - M$
(11)	$f \leftarrow t - i + 1$
(12)	$q \leftarrow GCD(f, N)$
(13)	$p \leftarrow {}^{N}/_{q}$
(14)	$d \leftarrow e^{-1}\ mod\ \varphi(N)$
(15)	**output:** p, q, d

Алгоритъмът извежда резултат единствено в случай, че на входа е подадено число N, за генерирането на което, е използван gBaseKleptoRSA2 с таблица S подадена като входен параметър при генерирането.

3.3 Сравнителен анализ и оценка на възможността за откриване.

Информацията по теми, свързани с клептографията е оскъдна на фона на всички останали дялове в криптографията. Тази проблематика все още не е задълбочено изследвана в сравнение с останалата теория, по въпроси, свързани със защитата на информация чрез криптиране. По тази причина сравнителните анализи на тема клептография, изложени в тази глава, са насочени към описани в публичното пространство клептографски алгоритми, атакуващи RSA.

Анализът на съществуващи клептографски алгоритми е насочен към такива, за които се твърди, че притежават свойства и функционалност, които ги правят практико приложими.

За да бъдат съпоставими тестовете необходими за извършване на анализ, те са изпълнявани на една и съща хардуерна компютърна конфигурация: CPU i5 10th generation (i5-1035G4) с 8 ядра, максимална честота 3.7GHz и RAM 16GB. При изпълнението на тестовете процесното натоварване бе в интервала от 60%-70%.

Всички клептографски алгоритми, които са предложени и създадени с цел атака на RSA се базират на предложеният от Йоунг и Юнг вариант и в повечето публикации реферират с него или кода им на изпълнение е много сходен с техният алгоритъм. По тази причина в дисертационния труд е описан единствено алгоритъма на Йоунг и Юнг и той е анализиран с цел оценка за практико-приложност. Последващият анализ може да се приема за обхващащ всички модификации и разработки на клептографски алгоритми за RSA, защото посочените в алгоритъма на Йоунг и Юнг редове на псевдо код, към които е насочен анализа са част и от повечето предлагани алгоритми в свободното интернет пространство, дори и в тези създадени с цел подобрение на отделни негови елементи.

Да разгледаме и анализираме следните редове от алгоритъмът на Йоунг и Юнг:

(1)	**random** $s \in z_{p-1}^{\times}$, $bitsize(s) \approx k/2$
(2)	$p \leftarrow hash\ (s)$
(3)	**if** $p\ is\ not\ prime$ **or** $GCD(e, p-1) \neq 1$
(4)	**goto (1)**

1) За да бъде реализиран се използва hash функция с резултат, чиято бинарна дължина трябва по условие да е равна или по-голяма от половината дължина на ключа (ред (1): $k/2$). Ако трябва да се генерира 2048 битов ключ, то съгласно това изискване трябва да използваме hash функция с резултат поне 1024 бита големина. Не са много на брой алгоритмите, предлагащи резултат с такава дължина. Можем да посочим само два SHA-3 и Skein, които освен, че покриват изискването за дължина на резултата от хеширане, са устойчиви и надеждни hash алгоритми. Това е другото изискване, което се посочва като условие в алгоритъма на Йоунг и Юнг, който анализираме. Ако се налага да се генерира по голям ключ, с 4096 бита например или по-голям, то в такъв случай предлагания клептографски алгоритъм няма как да функционира, защото не съществуват такива хеш функции, които да връщат резултат с такива размери (2048 бита и по-големи).

2) Съгласно ред (3), ако резултата от хеширане на произволните данни не е просто число или $GCD(e, p-1) \neq 1$, се връщаме за изпълнение на ред (1). Вероятността процеса сериозно да се забави в тази част на алгоритъма е голяма. Две причини са основание за това твърдение: бавното изпълнение на хеш алгоритмите, които се изисква да пресмятат хеш на данни с търсената дължина и второ, вероятността да се попадне на просто число в областта на тези големи числа не е достатъчно висока и това ще отнеме не малко итерации. Времето за генериране на ключ е важен показател. Ако необходимото време за генериране на ключ се отклонява сериозно от средно-статистическото време при стандартните алгоритми за генериране на RSA ключ (тези без клептография), това би било забележим признак за наличие на неправилно функциониране. Вследствие на такива подозрения може да се предполага наличие на клептография.

Да разгледаме друга част на предлагания от Йоунг и Юнг клептографски алгоритъм:

(7)	**solve:** $concat(c, RND)) = p.q + r$**,** $bitsize(q) \approx k/2$
(8)	**if** $q\ is\ not\ prime$ **or** $GCD(e, q-1) \neq 1$
(9)	**go to (1)**

На ред (7) е необходимо да се реши уравнение, при което от ляво е число, формирано от бинарна конкатенация на c и случайни данни получени на ред (5):

(5)	**random** $RND, bitsize(RND) \approx k/2$

Стойността на c е изключително важна, защото тя представлява резултат от криптирането на s (получено на ред (1)), с публичният ключ на атакуващия и както вече обърнахме внимание, чрез s е получен множителят p. В основната идея на атаката е, чрез възстановяване на s да се получи p и след това q. Ако погледнем отново ред (7), параметъра който може да бъде променян е само RND, а целта на решаване на уравнението е да се открие просто число q, което да е с приблизителна дължина $k/2$. За това са необходими итерации, за да получим такава стойност на r, така че: $q = \frac{concat(c,RND)) - r}{p}$.

За да не се забави алгоритъма прекалено много, се очаква стойността на r да е приемливо малка. Ако заради голяма стойност на r процеса се проточи, както вече изяснихме, това ще е забележим признак за наличие на неправилно функциониране. Ако пък r е с малка стойност и получим удовлетворяващ резултат за q, то това би означавало, че е много вероятно p да е много близко по стойност до q. Ако това е факт, то означава, че разликата $x - y$ (виж т. 1, глава 3, уравнение (29)) ще е с малка стойност, което респективно означава, че ще имаме съставно число с малко по стойност g, а както вече бе демонстрирано в първа точка на тази глава такова N ще е лесно разложимо.

Вторият показател за изследване на алгоритъма на Йоунг и Юнг е този за сигурност. Включва оценка на възможността за извличане на информация от тайния канал от лица не притежаващи ключа на атакуващия. Резултатът от анализа показа, че алгоритъма осигурява устойчив канал за защита на информацията, която изтича през него. Трудността на определяне на тайния параметър, който е послужил за формиране на p, е еквивалентна на решаване на задача за изчисляване на дискретен логаритъм.

От изложеното следва основният извод: варианти на алгоритъма на Йоунг и Юнг и всичките им модификации са практически не приложими за реализация в хардуерни модули (HSM, smartcard и др.) или софтуерни библиотеки. Основна причина за това ще е нелогично по-дългия период на генериране на ключ за RSA в сравнение с класическите криптографски алгоритми, в които не е включена клептографска атака.

За да бъде оценена степента за заплаха за устойчивостта на криптографски системи, базирани на RSA от използването на gBaseKleptoRSA2, бе направен анализ, включващ редица тестове приложени по подобие на анализа спрямо алгоритъма на Йоунг и Юнг. За да бъдат реализирани тестовете, беше създадено

софтуерно приложение, генериращо ключове за RSA, използващ gBaseKleptoRSA2 клептографски алгоритъм, което с цел постигане на обективен анализ бе изпълнявано на компютърната конфигурация приета за базова.

В първата част на анализа бяха извършени тестове с цел оценка на времето за генериране на ключове за RSA с големина 2048 бита. Средното време за генериране на ключ, което бе измерено е $\approx 1{,}98$ секунди. Това е съизмеримо време за генериране на ключове от този вид, сравнено с алгоритми, които не използват клептографска атака.

Вторият показател, по който бе изследван gBaseKleptoRSA2 е за сигурност на канала за изтичане на информация към атакуващия, за да може той да реализира клептографска атака за възстановяване на частния ключ. За да може да се определи еднозначно наличие на атака, е необходимо при изследването и анализа, да се възстанови изцяло или частично таблицата с тайни параметри (ключове) на атакуващия имплементирал gBaseKleptoRSA2.

Процесът на този анализ изисква да са направи оценка на двойки множители на модулни числа N, които участват във формиране на различен ключ. Например да се изследва p_i от един ключ с p_j от друг или двойка p_i с q_k $(k \neq i \neq j)$. Ако вследствие на такова изследване на двойки множители, се установи наличие на число w, за което:

$$w = GCD(x - r_x, y - r_y), \qquad w > 2^{32} \tag{35}$$

то може да се допусне, че такова w е потенциален член на S.

Използвайки посочените основни константи за реализация на gBaseKleptoRSA2 в предходната точка, при пълно комбиниране ще са ни необходими $877\,972\,608$ възможни комбинации на числа за изследване. Броят ключове, които трябва да генерираме, за да осигурим това количество числа е $438\,986\,304$. При средно време на генериране на ключ от $1{,}98$ секунди това прави $10\,060{,}10$ дни ($\approx 27{,}56$ години). Ако в процеса на генериране с използва суперкомпютър с 500 процесора Intel(R) Xeon(R) CPU E5-2660 v2 2.20GHz, то процеса на генериране би отнел $1\,100$ дни. Изразено като финансова инвестиция при средно $\$1\,250$ за cloud HSM (цена на IBM или AWS за месец) или $\$4{,}85$ на час (Microsoft Cloud HSM), е инвестиция от над 413 хиляди долара. С цел постигане на резултат от анализ за откриване на клептография gBaseKleptoRSA2 е необходимо и време за изчисления по формулата на уравнение (35) за всяко число от всичките $877\,972\,608$ числа (Y). При анализа за реализацията на търсене, използваме едно опорно число x, с което извършваме изчисления с всяко число от множеството на Y, за да открием $w \geq 2^{32}$. Средното време на изчисление, чрез използването на посочената базова конфигурация с процесор i5 е $7{,}39$ min, което се равнява на приблизително $12\,344{,}40$ години. Ако се използва паралелна форма на изчисления с помощта на Intel(R) Xeon(R) CPU E5-2660 (500 CPU), това би отнело $6{,}17$ години.

Увеличаването на интервала от стойности, в които се избират D_a и D_b води до увеличаване на трудността за детекция на клептография от този вид. Тестове показаха, че промяната на възможното множество от стойности на D_a и D_b от $1\,..\,97$ на $1\,..\,171$ и $d_a, d_b \in 64$ увеличи времето за разлагане на N (при размер 2048 бита) с по-малко от 10 секунди, но увеличението по отношение на минималното количество числа за изследване се увеличи на $9\,815\,261\,184$, което изисква над 11

пъти повече време (~ 69 години) за генериране на ключове и търсене на възможно w доказващо наличие на gBaseKleptoRSA2.

Други важни особености на използване на gBaseKleptoRSA2 е, че при генерирането на ключове с различна дължина може да се използва една и съща таблица S.

Ако обобщим то за предлаганата атака gBaseKleptoRSA2 можем да кажем:

- ***Приложима при различни дължини на ключа***;
- ***Трудна за установяване***.

Направените обобщения могат да са основа на твърдението, че gBaseKleptoRSA2 е изключително опасно средство за атака, което може да окаже сериозно влияние върху устойчивостта на функциониране на криптографски системи, базирани на RSA.

От казаното до тук възниква един важен въпрос: възможно ли е да съществува атака със сходен механизъм, която да функционира понастоящем в хардуерни средства или софтуерни библиотеки, които се използват в практиката. Това изисква анализ на вече съществуващи криптографски системи, насочен към процеса на генериране на ключове за RSA.

В края на тази точка на дисертационният труд е обърнато внимание на едно изследване на група чешки изследователи от университета Мазарик (Masaryk University) [57]. Публикуваните от тях резултатите са причина да се допусне, че в една част от изследваните технически средства и софтуерни библиотеки има наличие на слабости в процеса на генериране на ключове или съществуват форми на клептография. В обхвата на тяхното изследване попадат 38 средства за генериране на ключове. От тях 19 софтуерни библиотеки с отворен код, 3 комерсиални софтуерни продукта и 16 хардуерни средства.

Изложеното в тази заключителна част на дисертационният труд представя нагледно причините пораждащи необходимостта да се създаде средство за изследване и оценка на слабости при генериране на ключове за RSA. Това средство трябва да притежава функционалности за оценка и откриване на генерирани слаби ключове и за търсене на признаци за внедрена клептография.

3.4 Изводи.

От съдържанието на трета глава на дисертационният труд могат да се формулират следните изводи:

- Прилаганите стандарти в областта на криптографията, свързани със създаването на криптографски ключове за RSA се нуждаят от разширяване на условията, които трябва да бъдат изпълнени при генериране на двойки прости числа, които са предназначени за използване при формиране на ключ. Необходимо е включването на условие за оценка големината на отношението между простите числа p и q, които формират модулното число N.
- Математическите основи служещи за функциониране на RSA дават възможност при наличие на определени условия да бъде създаден механизъм за реализация на клептографска атака, която да е трудно откриваема. Това директно засяга в негативен аспект степента на

устойчивост на най-разпространения в световен мащаб алгоритъм използван в областта на публичната криптография, RSA.

- С цел постигане на достатъчни нива на устойчивост на криптографските системи, базирани на RSA е от съществена важност да бъдат разработени технически и софтуерни средства, позволяващи бързата, евтина и достъпна оценка за наличие на клептография.

ЗАКЛЮЧЕНИЕ

Развитието на публичната криптография е процес, при който темповете на поява на нови математически основи и алгоритми за криптиране не е висок и скоростта на внедряване в практическа експлоатация е изключително ниска. Това основно се дължи на факта, че се изисква време за проверка и изучаване на възможните и не дооценени предпоставки за създаване на математически, логически и алгоритмични уязвимости. Но с още по-бавни темпове се развиват и се появяват разработки, способстващи по-задълбоченото тестване и проверка за устойчивост на криптографските системи. Единствено такъв вид разработки са способни да дадат инструментариум за създаване на един по-качествен продукт под формата на криптографски средства за защита.

Развитието на технологиите е процес с високо темпо на нарастване, което създава предпоставка за увеличаващи се възможности за комуникация от тип машина-машина и комуникация между хора и води след себе си необходимост от съществуването на все по-големи масиви от чувствителна информация. Всичко това изисква използването и прилагането на надеждни и функционално сигурни системи за криптографска защита. Все по-големите количества информация, дигитализация и информационно развитие ще привлича все повече желаещи да се възползват от слабостите на системите с цел облагодетелстване.

Създаването на модел на решение, позволяващ подобряване оценката на качеството на генерираните ключове за RSA базирани криптографски системи, чрез бързи и детерминистични алгоритми за оценка на делимост на прости числа, води до повишаване на устойчивостта на функциониране на този вид системи.

Използването на базовите математически основи върху които е създадена съвременната публична криптография е предпоставка тя да бъде заложник на процесите свързани с бързо развиващите се изчислителни и квантови технологии. Риск за устойчивостта на криптографските системи, базирани на RSA съществува и в следствие и на появата на модели на решения, които предлагат възможност за повишаване на бързодействието при атаки на RSA. Оптимизирането на вече съществуващи алгоритми за атака, с помощта на такъв вид модели повишава бързодействието на атаката, което понижава устойчивостта на криптографските системи с публичен ключ.

Създаденият модел на клептографска атака, описан в дисертационния труд демонстрира реалната опасност за устойчивото функциониране на криптографски системи, използващи RSA. Базирайки се на резултатите от направените анализи и проверки може да се допусне, че сега съществуващите и прилагани в практиката стандарти, регулиращи процесите на генериране на RSA ключове вероятно трябва да включват и допълнителни условия на проверка с цел намаляване на вероятността за съществуване на клептография. Тази

необходимост произтича от факта, че този криптографски алгоритъм е най-разпространен и използван в практиката. Съгласно статистиката на организацията за прозрачност на цифрови сертификати към 2022 година неговото използване надхвърля 75%. Тази необходимост допълнително нараства и в следствие на многобройните свидетелства на установени уязвимости както в хардуерни решение за генериране на ключове за RSA, така също и в не малък брой софтуерни решения използвани за такива цели.

Основният извод, който може да бъде извлечен от дисертационния труд е, че публичната криптография ще продължи да се развива и прилага всеобхватно, но за да е максимално полезна на обществото, е необходимо да се намалят до минимум възможностите за нейното компрометиране, чрез използване на способи за пълноценна атака спрямо процесите на генериране на публични ключове.

Опита, познанията и данните натрупани при разработката на дисертационния труд ще бъдат използвани за целите на бъдеща работа свързана със създаването на модели на решения в следните области:

- Повишаване на устойчивостта и надеждността на генериране на ключове използвани в публичната криптография.
- Качествена и бърза оценка за наличие на внедрена клептография в системи, използващи публична криптография.
- Създаване на модели, които да послужат като основа, която да позволи създаването на алгоритмична база за работа в предстоящата пост-квантова ера.

Постигнати резултати

За постигане на поставената цел на дисертационния труд и изпълнение на свързаните задачи са постигнати следните научно-приложни и приложни резултати:

1. В първа точка на втора глава е предложен модел на решение, позволяващ повишаване на достоверността на резултата за оценка делимост на число като допълнение към най-широко прилагания в практиката алгоритъм, този на Милър-Рабин. Този модел създава възможност за постигане и на детерминистичен резултат без да се намалява бързодействието на изпълнение, която притежава алгоритъма на Милър-Рабин.

2. Във втора точка на втора глава е предложен нов модел на решаване на системи конгруентни уравнения, при който не съществуват ограничаващи условия, за да бъде модела изпълним. Липсата на ограничаващи условия го отличава от досега известните в практиката алгоритми, като този на „Китайската теорема“ и позволява постигане на решения без да е необходимо числата, с които се изчислява по модул да са взаимно прости. Използването на този модел в допълнение към алгоритъма за атака на RSA, предложен от Похлиг-Хелман повишава бързодействието на неговото изпълнение.

3. В трета глава на дисертационния труд е представен математически апарат и практическо решение, което позволява създаването на

клептографски алгоритъм за атака на RSA базирани криптографски системи. Направен е сравнителен анализи и оценка как клептографски алгоритми от този тип застрашават устойчивостта на функциониране на системи, използващи RSA. Тези резултати поставят въпрос за разглеждане, свързан с достатъчността на условията в състава на прилаганите стандарти отнасящи се до оценката на качеството на генериран RSA ключ.

Списък на свързаните с дисертацията публикации

[1] Stoianov, Nikolai, and Andrey Ivanov. "Public Key Generation Principles Impact Cybersecurity." Information & Security: An International Journal 47, no. 2 (2020): 249-260.

https://doi.org/10.11610/isij.4717

[2] Иванов А., Стоянов Н., Анализ на подходите за оценка на устойчивостта на криптографски алгоритми с публичен ключ, Десета международна научна конференция „Научните изследвания и инвестициите в технологични иновации – решаващ фактор за отбраната и сигурността“, ХЕМУС 2020, Пловдив

[3] Ivanov A.G., Stoianov N.T. (2020) New Approach of Solving Congruence Equation System. In: Dziech A., Mees W., Czyzewski A. (eds) Multimedia Communications, Services and Security. MCSS 2020. Communications in Computer and Information Science, vol 1284. Springer, Cham. https://doi.org/10.1007/978-3-030-59000-0_2