SPCSJ

SCIENTIFIC AND PRACTICAL CYBER SECURITY JOURNAL

VOL1 No2
DECEMBER 2017

ISSN 2587-4667

Scientific and Practical Cyber Security Journal (SPCSJ) 1(2):1-6
Scientific Cyber Security Association (SCSA), 2017 ISSN: 2587-4667

PASSWORDS AS A MEANS OF PROTECTION IN ORGANIZATIONS

O. Kovalchuk

Kyiv National University of Trade and Economics

ABSTRACT

Personal data of users is subject to damage, viruses, natural disasters, theft. Digital thieves are constantly looking for vulnerabilities that will allow them to steal valuable data. Attempts to steal information have different purposes: some scammers get money from bank accounts or credit cards, and others can sell information to a third party. The password is the simplest and cheapest way to authenticate. Restrictive password policies can cause some user actions, such as writing passwords, reusing them for different accounts, or sharing passwords with friends, can compromise security. The password security policy should balance directly between security and ease of use. The article proposes the worked out recommendations for the balance of security and user convenience.

KEYWORDS: password, security, organizations, protection, usability.

РЕЗЮМЕ

Персональные данные пользователей подвержены повреждениям, вирусам, стихийным бедствиям, кражам. Цифровые воры постоянно ищут уязвимости, которые позволят украсть ценные данные. Попытки украсть информацию имеют разные цели: одни мошенники - получают деньги с банковских счетов или созданных кредитных карт, а другие могут продавать информацию третьей стороне. Пароль является самым простым и дешевым способом аутентификации. Ограничительная политика паролей может привести к тому, что некоторые действия пользователей, например, записывание паролей, повторное использование их для разных учетных записей или совместное использование паролей с друзьями, могут поставить под угрозу безопасность. Политика безопасности паролей должна балансировать непосредственно между безопасностью и удобством использования. В статье предложены разработанные рекомендации для баланса безопасности и удобства пользователя.

Персональные данные пользователей подвержены повреждениям, вирусам, стихийным бедствиям, кражам. Цифровые воры постоянно ищут уязвимости, которые позволят украсть

ценные данные. Попытки украсть информацию имеют разные цели: одни мошенники - получают деньги с банковских счетов или созданных кредитных карт, а другие могут продавать информацию третьей стороне. Пароль является самым простым и дешевым способом аутентификации [1].

Пароли защищают личную информацию - информацию, которую мы не хотим передавать или не каждый должен знать. В нашей личной жизни это означает финансовую информацию, медицинские данные и частные документы. В профессиональном контексте это может охватывать все, что считается решающим для успеха организации: коммерческие секреты, финансовые данные, интеллектуальная собственность, списки клиентов и т.д. Создание пароля, по сути, привычное дело для пользователя, но далеко не каждый достаточно серьезно подходит к решению этого вопроса [2].

Рассмотрим разные вариации паролей со стороны безопасности и удобства использования.

1. 123456	**11.** 123123	**21.** mustang	**31.** 7777777	**41.** harley
2. password	**12.** baseball	**22.** 666666	**32.** flower	**42.** zxcvbnm
3. 12345678	**13.** abc123	23. qwertyuiop	**33.** qazwsx	**43.** asdfgh
4. qwerty	**14.** football	**24.** 123321	**34.** Jordan	**44.** buster
5. 123456789	**15.** monkey	**25.** 1234…890	**35.** Jennifer	**45.** andrew
6. 12345	**16.** letmein	**26.** princess	**36.** 123qwe	**46.** batman
7. 1234	**17.** shadow	**27.** superman	**37.** 121212	**47.** soccer
8. 111111	**18.** master	**28.** 270	**38.** killer	**48.** tigger
9. 1234567	**19.** 696969	**29.** 654321	**39.** trustno1	**49.** charlie
10. dragon	**20.** michael	**30.** 1qaz2wsx	**40.** hunter	**50.** robert

В таблице представлено 50 самых популярных паролей. Ни один из них не является достаточно хорошим или безопасным паролем, но они довольно простые для запоминания. По сути, это говорит нам, что при создании паролей предпочтение пользователя состоит в том, чтобы их было легко запомнить.

Хотя пароли являются жизненно важным компонентом системной безопасности, их можно легко взломать или сломать. Существует 5 проверенных способов:

1) *Запрашивание.* Является наиболее распространенным способом получения доступа к чьему-то паролю. То есть, для того чтобы узнать чей-то пароль, нужно просто попросить об этом (часто в связи с чем-то другим). Люди часто рассказывают свои пароли, совсем не беспокоясь о своей безопасности. Эта проблема не решится пока пользователи не начнут осознавать последствия таких своих действий.
2) *Угадывание.* Достаточно распространенный метод взлома. Большинство людей выбирают пароль, который легко запомнить, и самые простые - это те, которые связаны с вами как с личностью. Такие пароли легко угадать, зная минимальный объем информации о человеке, чей пароль запрашивается. Для предотвращения взлома данным способом, рекомендуем выбирать пароль, не имеющий отношение к пользователю как к человеку.

3) *Атака «грубого взлома».* Это обыкновенный перебор различных вариантов и комбинаций. Например, если у вас пароль «*sun*», программа попытается войти в систему, используя «aaa, aab, aac, aad ...sul, sum, *sun*». То есть – самые элементарные наборы цифр и букв. Эта программа способна в кратчайшие сроки перебрать огромное количество комбинаций. Единственное, что останавливает такую атаку - это сложность и более длинные пароли.
4) *Атака через общие слова.* Подобно атаке «грубого взлома», хакер пытается выполнить вход в систему, но уже используя список похожих слов, вместо комбинаций букв. Например, «sum, summer, summit, sump, *sun*».
5) *Атака по словарю.* Выполняется аналогично предыдущей атаке, единственное отличие заключается в том, что хакер теперь использует полностью весь словарь.

Первые два способа атак невозможно предотвратить только с помощью пароля, но есть возможность защитить систему от других форм атак [3]. Хакер обычно разрабатывает автоматизированный сценарий или программу, которые выполняют работу для него. Также система безопасности должна установить количество запросов на пароли, которые может сделать автоматическая программа - например, в секунду. Возможны разные варианты, но большинство веб-приложений не смогут обрабатывать более 100 запросов на вход в секунду. Это, по сути, и влияет на скорость взлома.

К примеру, если в качестве пароля используется:

- Любые цифровые комбинации, например, дата рождения - «15021986» - программа затратит около 2-х секунд на расшифровку;
- Именные пароли с маленькой буквы (anna, oleg) потребуют около 4-х секунд;
- Пароли, в которых используются имена с большой буквы (Anna, Oleg) – около 4-х минут;
- Более сложные комбинации с использованием цифр «1d2d3s4a8c» программа расшифрует за 4 дня;
- Пароли из серии «HSU5-BHJDa» будут расшифрованы через 12 лет;
- И комбинацию «J4fS<2» программа расшифрует через 219 лет.

Но означает ли это, что ИТ-отделы и компании по обеспечению безопасности правы, когда часто напоминают нам о том, что мы должны использовать сложные пароли, поскольку они являются более безопасными? Нет, это просто значит, что пароль с 6 символами не будет работать. Никто хочет запоминать пароль типа «J4fS<2», и, очевидно, что он будет записан в заметки.

Ограничительная политика паролей может привести к тому, что некоторые действия пользователей, например, записывание паролей, повторное использование их для разных учетных записей или совместное использование паролей с друзьями, могут поставить под угрозу безопасность. Другим нежелательным побочным эффектом определенной политики паролей является забывание паролей. На самом деле вред, причиненный пользователям после чрезмерно ограничительной политики паролей, может быть больше, чем вред, предотвращаемый этой политикой.

Соблюдение некоторых простых правил значительно облегчит жизнь пользователей, и решит проблему использования сложных паролей:

1) Прежде всего, в паролях нужно использовать слова, которые легко запоминаются, что-то простое и то, что можно быстро набрать (но это не должно быть связанно с пользователем как с личностью).

2) Также, когда в пароле используется не одно, а два простых слова, - это значительно увеличивает безопасность (для взлома пароля с одного слова программа потратит приблизительно 4 минуты, на пароль из двух - 2 месяца). Но, используя уже 3 слова, получается чрезвычайно безопасный пароль, для взлома которого понадобятся тысячи лет (в табл. 2 продемонстрированы примеры).

Type	Password	Method	Time	Security level
2 common word password	yellow bicycle	Common word	2 month	Low risk
3 common word password	tell the truth	Common word	2,537 years	**Secure forever**

3) И последнее, стоит пользователю придумать пароль с необычными словами, или же использовать больше 3-х коротких фраз, уровень защиты и время, которое понадобится для взлома, достаточно сильно возрастают (табл.3).

Type	Password	Method	Time	Security level
3 uncommon word password	pragmatic is realistic	Dictionary	39,637,200 years	**Secure forever**
5 uncommon word password	Du-bi-du-bi-dub	Brute-force	531,855,448,467 years	**Secure forever**

Исследования показывают такие результаты, так почему ИТ-департаменты заставляют нас ломать головы и пытаться запомнить очень сложные пароли? Если же присмотреться к офисным столам сотрудников любого департамента, с легкостью можно заметить, по крайней мере, несколько мониторов или столов, украшенных красочными пост-заметками с множеством паролей. Это признак того, что ИТ-администраторы ошиблись в своей политике паролей. И решение с точки зрения юзабилити заключается не в том, чтобы люди использовали заметки на столах или менеджеры паролей, а в том, чтобы сделать пароли более удобными.

Политика безопасности паролей должна балансировать непосредственно между безопасностью и удобством использованиях [4]. А система паролей будет более безопасной,

если пароль будет более удобным. Это объясняется как человеческим, таки компьютерным факторами. Для поддержания баланса мы рекомендуем:

1. *Обеспечить базовое обучение пользователей.* В первую очередь, этот принцип должен помочь сократить количество взломов под воздействием человеческого фактора (т.е. предотвратить угадывание и прямые запросы паролей). Беспрерывные нарушения безопасности происходят в результате человеческих ошибок или небрежности. В организациях нужно создать корпоративную культуру, которая подчеркивает безопасность компьютеров посредством учебных программ, предупреждающих о рисках небрежного использования паролей и неосторожного использования сетей, программ и устройств.
2. *Обратить внимание на энтропию (сложность) пароля*. Сложность пароля в компьютерной индустрии обычно измеряют в битах. Вместо количества попыток, которые необходимо предпринять для угадывания пароля, вычисляется логарифм по основанию 2 от этого числа, и полученное число называется количеством «битов энтропии» в пароле. Например, набор случайных символов, таких как «Tr0ub4dor&3», выглядит как супер безопасный пароль. Такой пароль имеет 2^{28}битов энтропии. Согласно формуле, при увеличении длины пароля на один бит количество возможных паролей удвоится, что сделает задачу атакующего в два раза сложнее. То есть, длинная цепочка запоминающихся для нас слов, например, «correct horse battery staple» имеет 2^{44} битов энтропии и является не только удобной, но и безопасной с технической стороны.
3. *Ограничить количество попыток входа.* В примерах выше устанавливали стандартное количество запросов на пароли, которые может сделать автоматическая программа - 100 запросов на вход в секунду. Но если администратор сократит количество попыток входа до 1 запроса на 5 секунд, и, более того, еще добавит штрафной период после 10 неудачных попыток, безопасность системы и ее взлома кардинально изменится, при этом не сильно повлияет на удобство использования (табл.4).

No of attacks	Password	Time	Security level
100 times per sec	yellow bicycle	2 month	Low risk
1 time every 5 sec	yellow bicycle	63 years	Secure
1 time every 5 sec with a 1 hour penalty period after 10 attempts	yellow bicycle	1,889 years	Secure forever

Очевидно, что пароль - всего лишь одна часть головоломки. Другие части – это политика безопасности, обучение пользователей и техническая составляющая обеспечивают гораздо более глобальную защиту в контролируемой корпоративной среде, чем пароли. Но в тех областях, где единственным способом контроля пользователей является ПИН или пароль, самое лучшее, что можно сделать, это знать о рисках безопасности и подобрать безопасный пароль.

Использованная литература

1. Dustin Van Der Haar, Basie Von Solms, "The poor man's biometric: Identifying cost-effective biometric system criteria for SMMEs", *IST-Africa Conference Proceedings 2014*, pp. 1-10, 2014.
2. M. Bellare, R. Canetti, and H. Krawczyk. A Modular Approach to the Design and Analysis of Authentication and Key Exchange Protocols. STOC ’98.
3. National Institute of Standards and Technology, "Digital Signature Standard," Federal Information Processing Standards Publication 186, May 1994.
4. Captcha development problems // Modern technics and technologies. 2015. № 7 [Electronic journal]. URL: http://technology.snauka.ru/en/2015/07/7577

Scientific and Practical Cyber Security Journal (SPCSJ) 1(2):7-14
Scientific Cyber Security Association (SCSA), 2017 ISSN: 2587-4667

ANALYSIS OF INCREASING HACKING AND CRACKING TECHNIQUES

Bilal Ahmad Kamal

Evergreen Public High school / University of Sargodha

ABSTRACT

In this work, the explanation is about how the computer's containing valuable information is being unsecured and the techniques to make it secure. This technique contains information on the tools and skills a hacker uses to infiltrate computer systems and networks. This work proposes the study of hacking; as the cost of hacking attacks continues to rise, many of the businesses have been forced to increase spending on network security. However, hackers have also developed new skills and techniques that allow them to break more complex systems. Hacking involves compromising the security of networks, breaking the security of application software's or creating malicious program such as viruses, threats, logic bombs, worms, etc[1]. This work describes the various techniques of hacking and cracking, also how they work. I proposed that the term "hacking and cracking" historically referred to constructive, clever technical work that was not necessarily related to computer systems. Today, however, hacking and hackers are most commonly associated with malicious programming attacks on the Internet and other networks.

INTRODUCTION

Hacking tricks can be divided into different categories elaborated below:

1. Trojan programs that share files via instant messenger.
2. Phishing
3. Fake Websites.
4. Spoofing
5. Spyware
6. Electronic Bulletin Boards
7. Information Brokers
8. Internet Public Records
9. Trojan Horses
10. Wormhole Attack Trojan programs that share files via instant messenger instant messaging allows file-sharing on a computer.

All present popular instant messengers have file sharing abilities, or allow users to have the above functionality by installing patches or plug-ins; this is also a major threat to present information security. These communication software also make it difficult for existing hack prevention method to prevent and control information security. Hackers use instant communication capability to plant Trojan program into an unsuspected program; the planted

program is a kind of remotely controlled hacking tool that can conceal itself and is unauthorized. The Trojan program is unknowingly executed, controlling the infected computer; it can read, delete, move and execute any file on the computer. The advantages of a hacker replacing remotely installed backdoor Trojan programs with instant messengers to access files are: When the victim gets online, the hacker will be informed. Thus, a hacker can track and access the infected computer, and incessantly steal user information. A hacker need not open a new port to perform transmissions; he can perform his operations through the already opened instant messenger port. Even if a computer uses dynamic IP addresses, its screen name doesn't change.

Hijacking and Impersonation

There are various ways through which a hacker can impersonate other users. The most commonly used method is eavesdropping on unsuspecting users to retrieve user accounts, passwords and other user related information. The theft of user account number and related information is a very serious problem in any instant messenger. For instance, a hacker after stealing a user's information impersonate the user; the user's contacts not knowing that the user's account has been hacked believe that the person they're talking to is the user, and are persuaded to execute certain programs or reveal confidential information. Hence, theft of user identity not only endangers a user but also surrounding users. Guarding against Internet security problems is presently the focus of future research; because without good protection, a computer can be easily attacked, causing major losses. Hackers wishing to obtain user accounts may do so with the help of Trojans designed to steal passwords. If an instant messenger client stores his/her password on his/her computer, then a hacker can send a Trojan program to the unsuspecting user. When the user executes the program, the program shall search for the user's password and send it to the hacker. There are several ways through which a Trojan program can send messages back to the hacker. The methods include instant messenger, IRC, emails, etc. Current four most popular instant messengers are AIM, Yahoo! Messenger, ICQ, and MSN Messenger, none of which encrypts its flow. Therefore, a hacker can use a man-in-the-middle attack to hijack a connection, then impersonate the hijacked user and participate in a chat-session.

An intrusion can be defined as an attempt to break into or to misuse a computer system for personal use. The word misuse; is a broad, and that can mean to something as severe as stealing confidential data to something as minor such as misusing your email system for spam, using your personal profiles, etc [1]. Today, both the Internet and corporate intranets are simply crawling with people from all walks of life that are continuously trying to test the security of various systems and networks for securing their data. Although the term “hacker” is in widespread use, the sense in which it is employed is generally incorrect. Popular media and entertainment providers have used it to describe anyone who tampers with a system, particularly involved in criminal activity. The hacker penetrates asystem remotely across the network. This journalistic misuse of the name upset many “traditional” hackers, who responded to the vilification of their good name by offering a new term for these individuals i.e. “crackers.”

Crackers are vandals and thieves whose sole purpose is unauthorized “cracking” into secure systems for personal gain

2. HACKING

In computer networking, hacking is any technical effort to manipulate the normal behaviour of network connections and connected systems. Hacking is an attitude and practice surrounding the use, consumption, and production of computer related work. Hacking uses Authorized or Unauthorized attempts to bypass the security mechanisms of an information systems or network. In simple words Hacking means finding out weaknesses in a computer or computer networks. The term "hacker" can mean two different things: 1. Someone who is very good at computer programming, networking, or other related computer functions and loves to share his knowledge with other people. 2. Someone who uses their expert computer skills and knowledge to gain unauthorized access to systems, corporations, governments, or networks for his personal use. Hacker seeks to understand computer, phone or other systems strictly for the satisfaction of having that knowledge. Hackers wonder how things work and have an incredible curiosity. Hackers will sometimes do questionable legal things, such as breaking into systems, but they generally will not cause harm once they break in. Contrast a hacker there is a term called cracker. Hacking is the practice of modifying the features of a computer system, in order to accomplish a goal outside of the creator's original purpose. The main goal of hacker is to steal the important or the secrete information or to destroy the enemies computer network. Computer hacking is the most popular form of hacking nowadays, especially in the field of computer security, but hacking exists in many other forms, such as phone hacking, brain hacking, server hacking, email hacking, etc. and it's not limited to either of them.

Hackers can be of three types i.e. 1. White hat hackers 2. Gray hat hackers 3. Black hat hackers 1. White hat hackers White hat hackers are the good guys who identify the security weakness of the system or network and inform the owner about them 2. Gray hat hackers A grey hat, in the hacking community, refers to a skilled hacker who is somewhere in between white and black hat hackers 3. Black hat hackers A black hat hacker is the villain or bad guy, who crash into victim’s security to steal information and destroy the victims security network

3.TECHNIQUES OF HACKING

These techniques comprises of either taking control over terminal or server to make it useless or to crash it. Following are the techniques used for hacking purpose explained as, 3.1 Denial of Service DoS attacks give hackers a way to bring down a network without gaining internal access. DoS attacks work by flooding the access routers with bogus traffic which can affect e-mail, TCP, packets. 3.2 Sniffing refers to the act of intercepting TCP packets. This interception can happen

through simple eavesdropping or something more sinister which modifies the packets. 3.3 Spoofing means pretending to be something you are not. In Internet terms it means pretending to be a different Internet address from the one you really have in order to gain something. 3.4 Viruses and Worms Viruses and worms are self-replicating programs or code fragments that attach themselves to other programs (viruses) or machines (worms). Both viruses and worms affect the networks by flooding them with huge amounts of bogus traffic, usually through e-mail. 3.5 Key loggers suppose, everything you type in the system is mailed to the hacker..!! It would be easy to track your password from that. Key loggers perform similar functionalities. So next time you type anything. Beware..!! 3.6 Social Engineering This was one of the oldest tricks for hacking. If the hacker try to convince user that you are a legitimate person from the system and needs your password for the continuation of the service or some maintenance. But it won't work because it is an old technique. 3.7 Fake Messengers In this hacking technique, attacker send the fake messages so that the user can fill his own information like login id, password , etc[4],[5].

4. CRACKING

Cracker is the common term used to describe a malicious hacker. Crackers get into all kinds of mischief, including breaking or "cracking" copy protection on software programs, breaking into systems and causing harm, changing data, or stealing. Hackers regard crackers as a less educated group of individuals that cannot truly create their own work, and simply steal other people's work to cause mischief, or for personal gain. Crackers break into or crack systems with malicious intent. They are out for personal gain: fame, profit, and even revenge. They modify, delete, and steal secret information, often making other people miserable

5 TECHINIQUES OF CRACKING

There are three basic types of password cracking techniques that can be explained below,

1. Dictionary Cracker can run a file of words against user accounts, and if the password is found to be simple word, it can be found pretty quickly.
2. Hybrid We know that the user utilize common method to change passwords is to add a number or symbol to the end. A hybrid attack works like a dictionary attack, but adds simple numbers or symbols to the password attempt.
3. Brute force This technique of hacking is complex one and time-consuming, but comprehensive way to crack a password. Every combination of character is tried until the password is broke.

4. PROPOSED IDEA There are a few variations on the types of attacks that successfully employ hacking. Although some are relatively dated, others are very pertinent to current security concerns. Hacking consists of several steps, which I will briefly outline here, then explain in detail

5. First, the target host is chosen. Next, a pattern of trust is discovered, along with a trusted host. The trusted host is then disabled, and the target's TCP sequence numbers are sampled. The trusted host is impersonated, the sequence numbers guessed, and a connection attempt is made to

a service that only requires address-based authentication. If successful, the attacker executes a simple command to leave a backdoor. Details of an Attack Hacking in brief consists of several steps; 1. Selecting a target host (or victim). 2. The trust relationships are reviewed to identity a host that has a "trust" relationship with the target host 3. The trusted host is then disabled and the target's TCP sequence number are sampled. 4. The trusted host is then impersonated, the sequence number forged (after being calculated). 5. A connection attempt is made to a service that only requires address-based Authentication (no user id or password).
6.IF a successful connection is made, the attacker executes a simple command to leave a backdoor.

7. DIFFERENCE BETWEEN HACKING AND CRACKING

The main difference between these two techniques is "Hacking builds things" and "Cracking breaks them". Hacking and cracking are the two different forms of Internet and computer related privacy, usually hazardous. In this article, I'll be discussing the differences between hacking, and cracking[8]. They are two completely different things, but people usually get confused between the two, they both end with a similar sound 'Hacking' and they're both malicious forms of cyber activity. In this article, I'll be talking about the difference between hacking, and cracking. A hacker is someone who seeks and exploits weaknesses in a computer system or computer network. Hackers may be motivated by a multiple of reasons, such as profit, protest, or challenge. Malicious attacks on computer networks are officially known as cracking, while hacking truly applies only to activities having good intentions. Most non-technical people fail to make this distinction. Besides all these, it's extremely common to see the term "hack" misused and be applied to cracks as well

8. CONCLUSION

In this work, at last I want to conclude that, how the attacker affects the security which tremendously affect our growing environment. Many security experts are predicting a shift for hacking in which hackers can exploit a weakness in a particular service to send and receive information under false identities. As Security professionals, we must remain current with the Operating Systems that we use in our day to day activities. A steady stream of changes and new challenges is assured as the hacker community continues to seek out vulnerabilities and weaknesses in our systems and our networks. Understanding how and why attacks are used, combined with a few simple prevention techniques, can help protect your network from hacking.

Our main goal in this paper is to show how these two techniques i.e. hacking and cracking are performed in various ways and how the attacker can easily attack the users system using these two techniques.

SOLUTIONS

Social responses One strategy for combating phishing is to train people to recognise phishing attempts and to deal with them . Education can be promising, especially where training provides direct feedback. People can take steps to avoid phishing attempts by slightly modifying their browsing habits. When contacted about an account needing to be "verified" (or any other topic used by phishers), it is a sensible precaution to contact the company from which the email apparently originates to check that the email is legitimate. Alternatively, the address that the individual knows is the company's genuine website can be typed into the address bar of the browser, rather than trusting any hyperlinks in the suspected phishing message. Technical responses Anti-phishing measures have been implemented as features embedded in browsers, as extensions or toolbars for browsers, and as part of website login procedures. The following are some of the main approaches to the problem. Helping to identify legitimate sites Since phishing is based on impersonation, preventing it depend on some reliable way to determine a website's real identity. For example, some anti-phishing toolbars display the domain name for the visited website. The pet-name extension for Fire-fox lets users type in their own labels for websites, so they can later recognize when they have returned to the site. If the site is suspect, then the software may either warn the user or block the site outright. Fake Web sites Fake bank websites stealing account numbers and passwords have become increasingly common with the growth of online financial transactions. Hence, when using online banking, we should take precautions like using a secure encrypted customer's certificate, surf the net following the correct procedure, etc. First, the scammers create a similar website homepage; then they send out e-mails with enticing messages to attract visitors. They may also use fake links to link internet surfers to their website. Next, the fake website tricks the visitors into entering their personal information, credit card information or online banking account number and passwords. After obtaining a user's information, the scammers can use the information to drain the bank accounts, shop online or create fake credit cards and other similar crimes. Usually, there will be a quick search option on these fake websites, luring users to enter their account number and password. When a user enters their account number and password, the website will respond with a message stating that the server is under maintenance. Hence, we must observe the following when using online banking: Observe the correct procedure for entering a banking website. Do not use links resulting from searches or links on other websites. Online banking certifications are currently the most effective security safeguard measure. Do not easily trust e-mails, phone calls, and short messages, etc. that asks for your account number and passwords.
Solutions Internet Explorer 7 and Fire-fox 2 both have sophisticated filters that can detect most fake websites. Here are some other clues that might give away a fake: • Look for evidence of a real-world presence: an address, a phone number, an email contact. If in doubt, send an email, make a phone call or write a letter to establish whether they really exist. • The website's address is different from what you are used to, perhaps there are extra characters or words in it or it uses a completely different name or no name at all, just numbers. • Right-clicking on a hyperlink and selecting "Properties" should reveal a link's true destination - beware if this is different from what is displayed in the email. • Even though you are asked to enter private information there is NO padlock in the browser window or 'https://' at the beginning of the web address to signify

that it is using a secure link and that the site is what it says it is. • A request for personal information such as user name, password or other security details IN FULL, when you are normally only asked for SOME of 49 them. • Although rare, it is possible for your computer to be corrupted by viruses in such a way that you can type a legitimate website address into your browser and still end up at a fake site. This problem is known as 'pharming'. Check the address in your browser's address bar after you arrive at a website to make sure it matches the address you typed. Subtle changes ('eebay' instead of 'ebay' for example) may indicate that your computer is a victim of a pharming attack. Pharming Similar in nature to phishing, Pharming (pronounced farming) is a Hacker's attack aiming to redirect a website's traffic to another, bogus website. Pharming can be conducted either by changing the hosts file on a victim's computer or by exploitation of a vulnerability in DNS server software. DNS servers are computers responsible for resolving Internet names into their real addresses - they are the "signposts" of the Internet. Compromised DNS servers are sometimes referred to as "poisoned". The term pharming is a word play on farming and phishing. The term phishing refers to social engineering attacks to obtain access credentials such as user names and passwords. In recent years pharming has been used to steal identity information. Pharming has become of major concern to businesses hosting ecommerce and online banking websites. Spoofing A technique used to gain unauthorized access to computers, whereby the intruder sends messages to a computer with an IP address indicating that the message is coming from a trusted host. To engage in IP spoofing, a hacker must first use a variety of techniques to find an IP address of a trusted host and then modify the packet headers so that it appears that the packets are coming from that host. A closely interconnected and often confused term with phishing and pharming is spoofing. A "spoofer", in Internet terms, is defined generally as the "cracker" who alters, or "forges", an e-mail address, pretending to originate a message from a different source address than that which he or she truly has. There are many ways an attacker may do this, and there are many types of attacks. The attacker may do this to gain access to a secured site that would accept the "hijacked" address as one of few permissible addresses, or more maliciously, the reason may be to hide the source of any type of attack. Email spoofing is often an attempt to trick the user into making a damaging statement or releasing sensitive information (such as passwords). Spoofing Attacks Techniques Spoofing attacks can be divided into different categories, some of which are elaborated below: Man-in-the-middle attack and internet protocol spoofing An example from cryptography is the man-in-the-middle attack, in which an attacker spoofs Alice into believing they're Bob, and spoofs Bob into believing they're Alice, thus gaining access to all messages in both directions without the trouble of any.

REFERENCES

1. Hacking: The Basics,Zachary WilsonApril 4, 2001Updated by Martin Poulin, GSEC, GCWN, GCIH, CISSPJune 27, 2006
2. The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue: The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13], San Francisco State University, CA, U.S.A., March 2013 Doi: 10.7321/jscse.v3.n3.74 e-ISSN: 2251-7545

3. Analysis of Increasing Malwares and Cyber Crimes Using Economic Approach
4. 2010 International Journal of Computer Applications (0975 – 8887) Volume 1 – No. 10,Ethical Hacking Procedure Certified Ethical Hacking Ethical Hacking : Future
5. All Types Of Hacking Techniques, http://hackerhubz.blogspot.in/2009/10/all-types-of-hacking-techniques.html
6. Stevens, W. Richard. TCP/IP Illustrated Volume I: The Protocols. Boston, Mass. : Addison-Wesley, 2004.
7. Wright G.R., Stevens, W. R. TCP/IP Illustrated Volume II: The Implementation. Boston, Mass. : Addison-Wesley, 1995.
8. Comer, Douglas E. Internetworking with TCP/IP Volume I: Principles, Protocols and Architecture. Upper Saddle River, New Jersey. : Prentice Hall, 1995
9. Practical Hacking Techniques and Countermeasures by Mark D. Spivey, CISSP
10. Is Ethical Hacking Ethical?,Danish Jamil et al. / International Journal of Engineering Science and Technology (IJEST)
11. Teaching ethical hacking in information security curriculum: A case study,Global Engineering Education Conference (EDUCON), 2013 IEEE,13-15 March 2013.

Scientific and Practical Cyber Security Journal (SPCSJ) 1(2):14-20
Scientific Cyber Security Association (SCSA), 2017 ISSN: 2587-4667

MERKLE WITH QUANTUM TRNG

A.Gagnidze, M.Iavich, G. Iashvili
Scientific Cyber Security Association

ABSTRACT:

Scientists are actively working on the development of quantum computers. Traditional cryptosystem systems that are used in practice are vulnerable to attacks by quantum computers. The security of these systems is based on the problem of factoring large numbers and calculating discrete logarithms. Active work is being conducted to create RSA alternatives, which are protected from attacks by a quantum computer. One of the proposed alternatives are hash based digital signature systems. The security of these crypto systems is based on the collision resistance of hash functions, which they use. In the article is proposed the novel version of Merkle crypto system. The system uses TRNG based on the state of qubits. The system is secure, because we do not change the principle of the crypto system, but only integrate TRNG, to reduce the size of the signature key. TRNG is completely safe; It is based on the state of qubits, which are real random number.

KEYWORDS: Merkle, quantum, TRNG, crypto, security.

РЕЗЮМЕ:

Ученые активно работают над разработкой и усовершенствованием квантовых компьютеров. Традиционные криптосистемы, которые используются в практике уязвимы к атакам квантовых компьютеров. Безопасность данных систем основана на проблеме факторизации больших чисел и вычислении дискретных логарифмов. Ведется активная над созданием альтернатив RSA, которые защищены от атак квантового компьютера. Одной из предложенных альтернативой являются системы электронной подписи, основанные на хешировании. Безопасность данных крипто систем основывается на стойкости к коллизиям хеш функций, которые они используют.

В статье предложена новый вариант крипто системы Merkle. Система использует TRNG основанный на состояниях кубитов. Система является безопасной, т.к. мы не меняем принцип работы крипто системы, а только вставляем TRNG, для уменьшения размера ключа подписи.

TRNG является полностью безопасным, т.к. он основан на состоянии кубитов, которое является реальным случайным числом.

Новая криптосистема, основанная на квантовом генераторе случайных чисел. Ученые активно работают над разработкой и усовершенствованием квантовых компьютеров. Традиционные системы криптосистемы, которые используются в практике уязвимы к атакам квантовых компьютеров. Безопасность данных систем основана на проблеме факторизации больших чисел и вычислении дискретных логарифмов.

Ведется активная над созданием альтернатив RSA, которые защищены от атак квантового компьютера. Одной из предложенных альтернативой являются системы электронной подписи основанные на хешировании. Безопасность данных крипто систем основывается на стойкости к коллизиям хеш функций, которые они используют[1].

Схемы одноразовой подписи

Были предложены одноразовые схемы электронной подписи. Была предложена схема одноразовой подписи Лэмпорта (Lamport–Diffie one-time signature scheme), данная схема является электронной подписью основанной на хешировании и представляет альтернативу для пост квантовой эпохи[2]. В данной схеме генерация ключа и генерация подписи довольно эффективна, но размер подписи является довольно большим. Для уменьшения подписи была предложена схема одноразовой подписи Винтерница (Winternitz one-time signature scheme). В данной схеме одной строчкой ключа подписываются одновременно несколько битов хешированного сообщения, этим существенно уменьшается длина подписи.

Схемы одноразовой подписи очень неудобны в использовании, т.к. для подписи каждого сообщения нужно использовать разную пару ключей. Была предложена крипто система Меркле, для решения этой проблемы. В данной системе используется бинарное дерево, чтобы заменить большое количество ключей верификации одним открытым ключом, корнем бинарного дерева. Данная криптосистема использует схему одноразовой подписи Лэмфпорта или Винтерница и криптографическую хеш функцию:

$h:\{0,1\}^* \rightarrow \{0,1\}^n$

Генерация ключа: Выберается длина дерева H>=2, одним открытым ключом можно подписать 2^H документов. Генерируются 2^H пар ключей подписи и верификации X_i, Y_i, $0<=i<=2^H$. X_i- ключ подписи**,** Y_i- ключ верификации. Вычисляются $h(Y_i)$ и используются как листья дерева. Каждый узел дерева является хешированием объединения его детей.

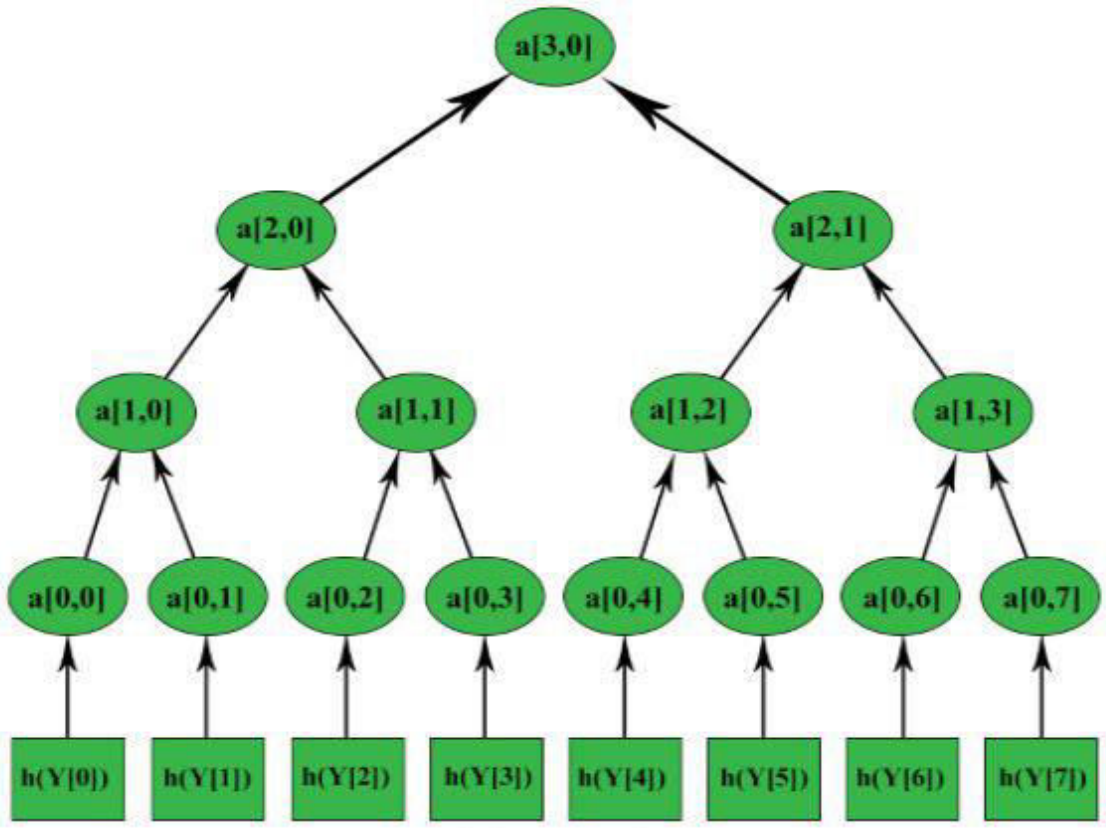


Рис. 1

На рисунке Рис. 1 показано дерево с H=3; a[i,j] узлы дерева; a[1,0]=h(a[0,0] || a[0,1]). Корень дерева является открытым ключом крипто системы - pub. Для генерации открытого ключа нужно высчитать $\mathbf{2^H}$ пар одноразовых ключей и использовать функцию h $\mathbf{2^{H+1}-1}$ раз.

Генерация подписи:

Для подписи сообщения m произвольного размера мы переводим его в размер n с помощью функции хеширования h(m) = hash, и генерируем одноразовую подпись [2], используя любой одноразовый ключ X_{any}, подпись документа будет объединением: одноразовой подписи, одноразового ключа верификации Y_{any} , индекса any и всех братских узлы $auth_i$ по отношению к Y_{any}.

Signature= (sig||any|| Y_{any}||$auth_0$,…,$auth_{H-1}$)

Верификация подписи:

Для верификации подписи мы проверяем одноразовую подпись sig с помощью Y_{any}, если она верна высчитываем все узлы a[i,j], используя $auth_i$, any и Y_{any}. Сравниваем последний узел-корень дерева с открытым ключем, если они равны то подпись верна.

Интеграция PRNG:

Для генерации открытого ключа нужно высчитать и хранить $\mathbf{2^H}$ пар одноразовых ключей. Хранить такое количество информации не эффективно в практике. Для того чтобы сохранить место было предложено использовать псевдо генератор случайных чисел PRNG [3]. При использовании PRNG достаточно хранить только семя генератора и использовать его для генерации одноразовых ключей. Нужно высчитывать одноразовые ключи дважды: один раз в стадии генерации ключей и второй раз в стадии подписи сообщения. PRNG получает семя длины n и выдает новое семя и случайное число длины n.

$PRNG : \{0, 1\}^n \rightarrow : \{0, 1\}^n \times \{0, 1\}^n$

Генерация ключа используя PRNG:

Выбирается семя s_0 длины n случайным образом, с помощью s_i мы вырабатываем sot_i, следующим образом:

$PRNG(s_i) = (sot_i , s_{i+1}) \quad 0 \leq i < 2^H$

sot_i каждый раз меняется при запуске PRNG. Для вычисления ключа X_i, достаточно знать только s_i. Работа PRNG показана на рисунке Рис. 2.

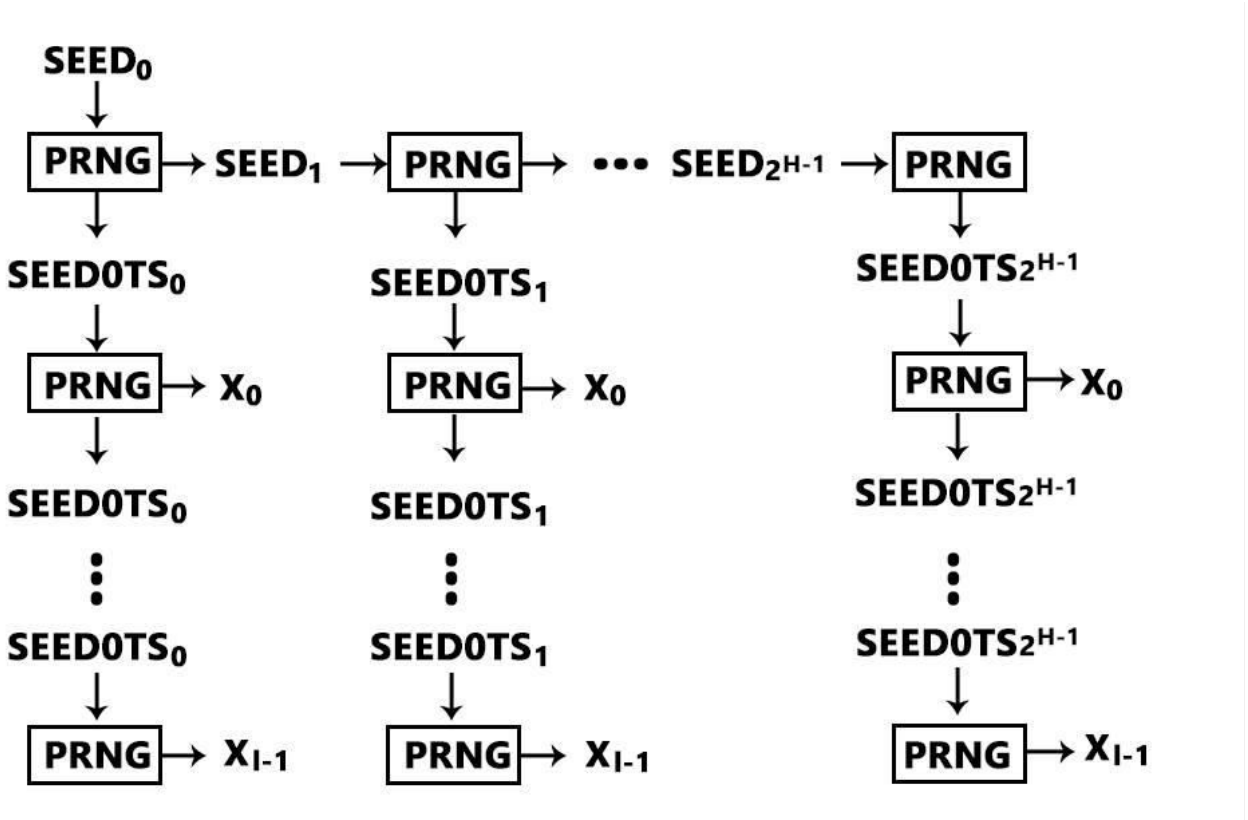


Рис. 2.

Подпись и верификация происходит аналогично как в стандартном варианте крипто системы Меркле.

Квантовые компьютеры способны взломать PRNG, которые считались безопасными против атак классических компьютеров[4]. Показана полиномиальная квантовая атака времени на PRNG Blum-Micali, который считается безопасным от угроз со стороны стандартных компьютеров. Эта атака использует алгоритм Гровера вместе с квантовым дискретным логарифмом, и способна восстанавливать значения на выходе генератора при данной атаке. Такие атаки представляют угрозу взлома PRNG, используемых во многих криптосистемах реального мира. Как мы видим крипто система Меркле с встроенным PRNG может быть уязвима к атакам квантовых компьютеров. Мы предлагаем использовать истинный генератор случайных чисел основанный на кубитах, TRNG.

Для построения данного TRNG рассмотрим квантовые состояния и кубиты.

Квантовый TRNG:

Рассмотрим систему с одним кубитом. Квантовое состояние кубита обозначается как:

$\alpha|0>+\beta|1>$, где α и β комплексные числа; $|\alpha|^2+|\beta|^2=1$

$|0>$ - земное состояние кубита, $|1>$- возбужденное состояние кубита

Данный кубит находится в состоянии $|0>$ с вероятностью α^2 и аналогично в состоянии $|1>$ с вероятностью β^2.

При измерении кубита он оказывается в одном из двух состояний с вероятностью 1.

Рассмотрим систему с двумя кубитами. Квантовое состояние двух кубитов обозначается как:

$\alpha_{00}|00>+\alpha_{01}|01>+\alpha_{10}|10>+\alpha_{11}|11>$

где α_i комплексные числа; $\sum|\alpha_i|^2=1$.

Свяжем эти кубиты с помощью парадокса Белла (Bell state), тогда квантовое состояние этих кубитов обозначается как:

$1/2^{1/2}|00>+1/2^{1/2}|11>$, т.е. при измерении данного кубита он окажется в состоянии $|0>$ с вероятностью ½ и аналогично в состоянии $|1>$ с вероятностью ½. При измерении второго кубита он окажется в том же состоянии, в котором был первый кубит при измерении с вероятностью 1.

Измеряя n кубитов можно получить истинное число размера n.

Генерация ключа с помощью квантового TRNG:

Выбирается длина дерева H>=2, одним открытым ключом можно подписать 2^H документов. Между двумя кубитами устанавливается связь с помощью парадокса Белла. Берутся 2^H пар таких кубитов q_i и b_i ;каждый q_i и b_i состоит из n кубитов. $0<=i<=2^H$; Измеряем 2^H*n кубитов q_i получаем 2^H ключей подписи X_i и вычисляем ключи верификации Y_i,. Вычисляются $h(Y_i)$ и используются как листья дерева.

Подпись и верификация сообщения:

Для подписи сообщения m произвольного размера мы переводим его в размер n с помощью функции хеширования h(m) = hash, измеряем множество состоящее из n кубитов , b_{any} = qany=X_{any} с вероятностью равной 1. Используя одноразовый ключ подписи X_{any} генерируем одноразовую подпись, подпись документа будет объединением: одноразовой подписи, одноразового ключа верификации Y_{any} , индекса any и всех братских узлы $auth_i$ по отношению к Y_{any}.

Signature= $(sig\|any\|Y_{any}\|auth_0,\ldots,auth_{H-1})$

Верификации сообщения происходит аналогично как в стандартной системе Меркле.

Безопастность системы с встроенным квантовым PRNG.

Система является безопасной, т.к. мы не меняем принцип работы крипто системы, а только вставляем TRNG, для уменьшения размера ключа подписи. TRNG является полностью безопасным, т.к. он основан на состоянии кубитов, которое является реальным случайным числом.

1. Гагнидзе А.Г., Явич М.П., Иашвили Г.Ю. Пост-квантовые криптосистемы // Современные научные исследования и инновации. 2016. № 5 [Электронный ресурс]. URL: http://web.snauka.ru/issues/2016/05/67264

2. Gagnidze. A. G. , Iavich. M. P. , Inasaridze. N. K. , Iashvili. G. I. , Analysis of one-time signature schemes// Scientific & practical cyber security journal (SPCSJ) № 1.Electronic journal]. URL: http://journal.scsa.ge/issues/2017/09/455

3. Buchmann, J., Coronado, C., Dahmen, E., Döring, M., Klintsevich, E.: CMSS – an improved Merkle signature scheme. In Progress in Cryptology - INDOCRYPT 2006, LNCS 4329, pages 349–363. Springer-Verlag, 2006.

4. Change GUEDES, E., DE ASSIS, F., & LULA, B. (2013). Quantum attacks on pseudorandom generators. Mathematical Structures in Computer Science, 23(3), 608-634. doi:10.1017/S0960129512000825

Scientific and Practical Cyber Security Journal (SPCSJ) 1(2):21-27
Scientific Cyber Security Association (SCSA), 2017 ISSN: 2587-4667

FORENSIC-CHAIN: ETHEREUM BLOCKCHAIN BASED DIGITAL FORENSICS CHAIN OF CUSTODY

Auqib Hamid Lone, Roohie Naaz Mir

Department of Computer Science and Engineering NIT Srinagar, Jammu and Kashmir 190006

ABSTRACT

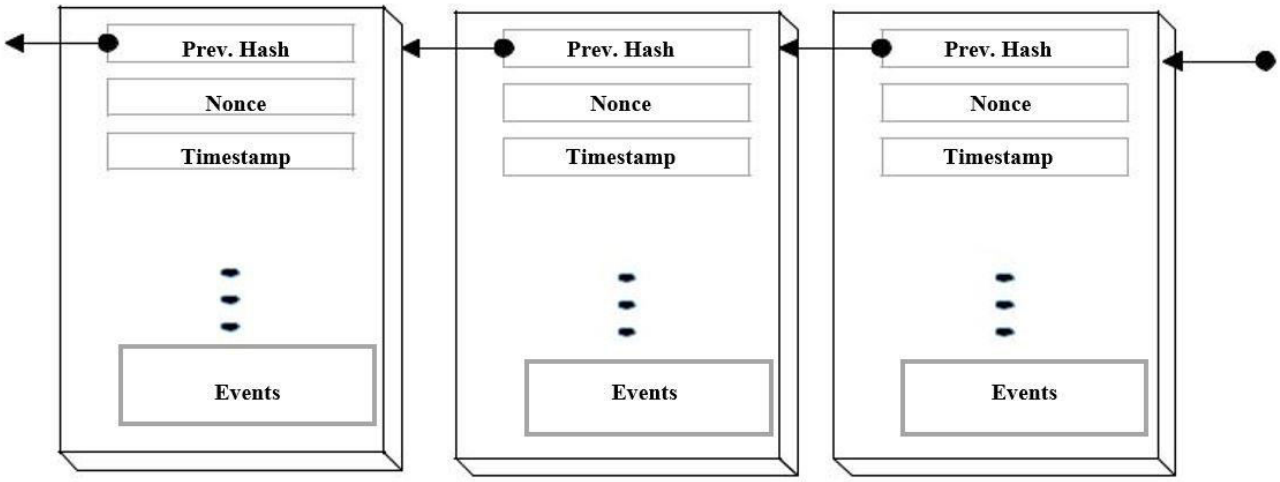
Digital evidence plays an important role in cyber crime investigation, as it is used to link persons with criminal activities. Thus it is of extreme importance to guarantee integrity, authenticity, and auditability of digital evidence as it moves along different levels of hierarchy in chain of custody during cyber crime investigation. Blockchain technology's capability of enabling comprehensive view of transactions (events/actions) back to origination provides enormous promise for the forensic community. In this research we proposed to use a blockchain that can be leveraged for forensic applications in particular bringing integrity and tamper resistance to digital forensics chain of custody.

KEYWORDS: Blockchain, Ethereum, Chain of Custody, Smart Contracts.

INTRODUCTION

In today's digital world, with rapid increase in cyber crimes, the importance of digital evidence is also growing for provenance of persons linkage with cyber crimes. Digital evidence comes with its own unique challenges related to chain of custody. Chain of custody can be defined as a process used to maintain and document the chronological history of han-dling digital evidence [1]. In digital forensics evidence passes through different levels of hierarchy i.e from first responder to higher authorities responsible for handling cyber crime investigation. During this passage of digital evidence there is always higher degree of integrity violation and repudiation. As a matter of fact, the need of the hour is to have a system that guarantees transparency, authenticity, security and auditability. Blockchain in its simplicity is a series of connected data structure called blocks, which contains or tracks everything that happens on some distributed systems on a peer to peer to network [2]. Each block is linked to and depends on previous block forming a chain, resulting in an append only system: a permanent and irreversible history that can be used as a real time audit trail by any participant to verify the accuracy of the records by simply reviewing data itself. Ethereum is a

blockchain with built-in Turing-complete programmming language, giving users power to write smart contracts, de-centralized applications where users define their own arbitary rules for



ownership, transaction formats and state transition functions [3]. Blockchain by design guarantees transparency, authenticity, security and auditability and thus making it best fit for maintaining and tracing chain of custody for forensic applications. The motivation behind using ethereum blockchain smart contracts is that they provide more power in terms of Turing-completeness, value-awareness, blockchain-awareness.

Fig. 1. Blockchain and state.

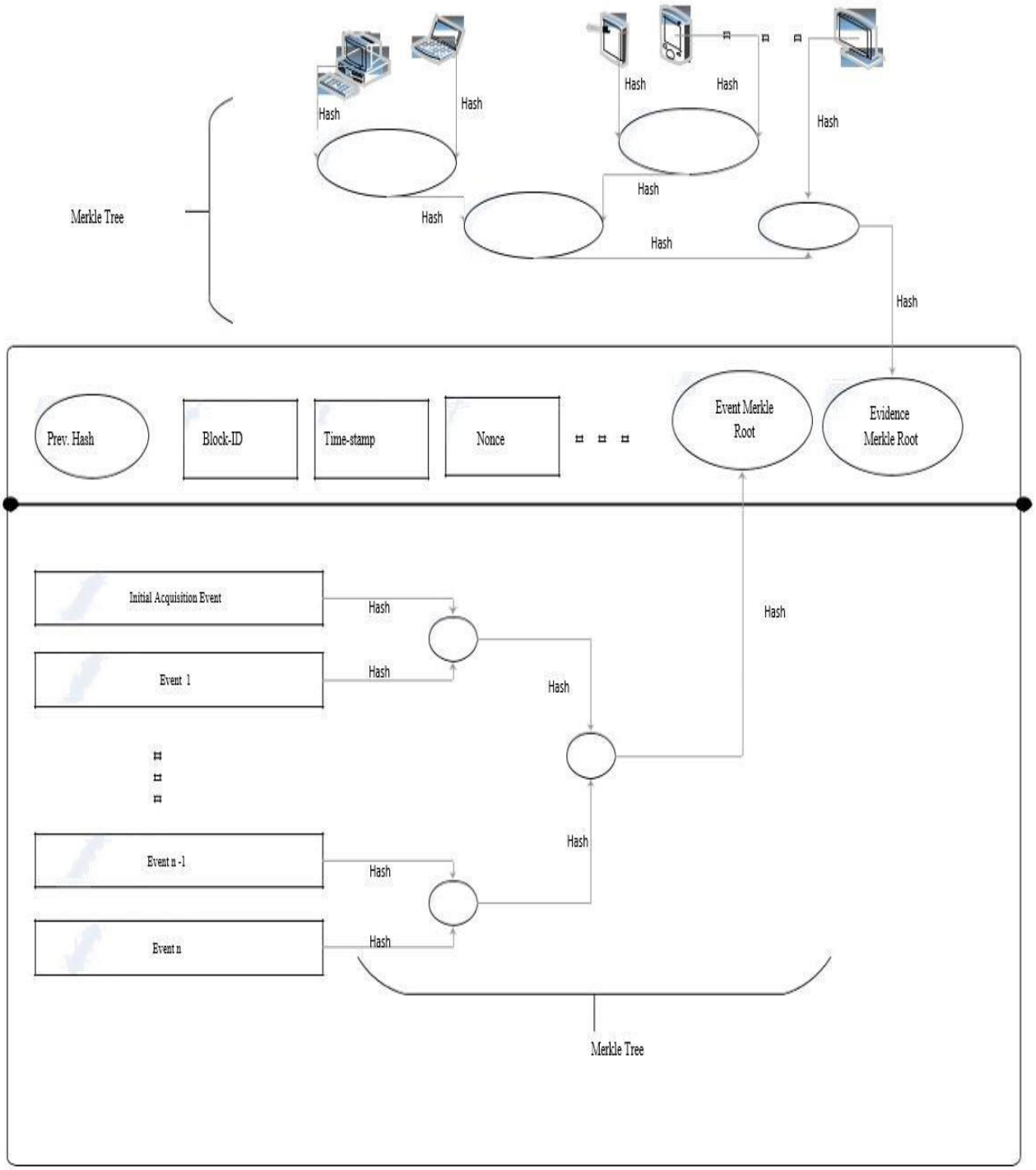
PROPOSED FORENSIC-CHAIN MODEL

Forensic-Chain is a blockchain based solution for maintain-ing and tracing digital forensics chain of custody. Blockchain is a data structure that allows to create a digital ledger for recording and storing transactions (events/records) shared by all participating parties over a distributed network of computers. Blockchain makes use of cryptography for protecting the process of recording and storing transactions (events/records) that happen within the network, creating unimpeachable audit trail.

In relation to chain of custody, the blockchain's capability specifically in combination with cryptographic hashing and encryption could potentially create documentation pertaining to access to evidence that is tamper- proof [4], [5] . The evidence that is to be preserved is first encrypted securely and have a blockchain capability added on.The encrypted data would be accessible only to desired party on the blockchain but would simultaneously record the time,date and possibly user-ID of the accessing party and add it to the unalterable record in blockchain all done automatically through smart contract The blockchain itself can be read via a special function in a way that is similar to how the bitcoin blockchain can be decoded. This functionality of blockchain allows courts

and associated personnel the ability to examine historical chain of custody without accessing data itself.

Actual implementation is as follows: The blockchain's first entry i.e genesis block com-prises of



initial hash of the data such as time,date and location of initial acquisition as shown in figure 2.

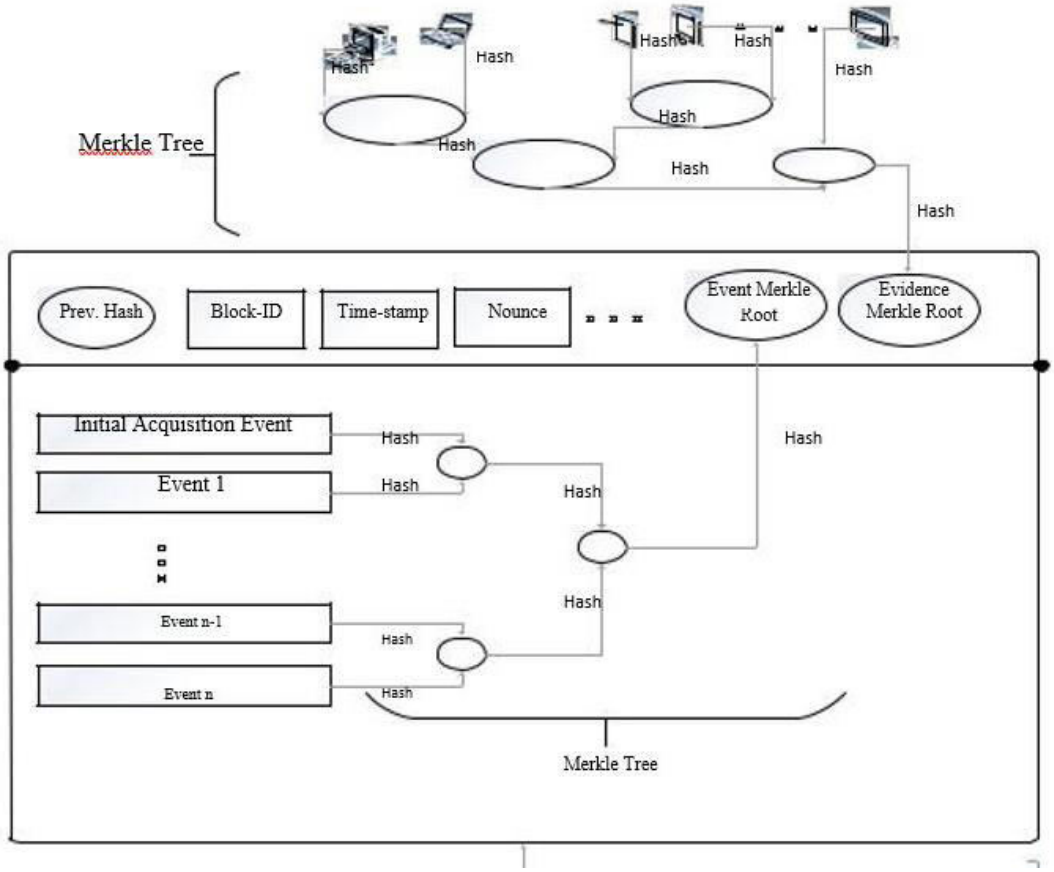
Fig. 2. Genesis Block of Forensic Chain Model

1. Subsequent accesses made possible only via a special program (like the one used to read a bitcoin blockchain), will add additaional blockchain entries each time access or transfer of evidence occurs. In simpler terms during subsequent accesses of evidence a new, non-repudiable, irreversible, cryptographically secure block gets added to blockchain based chain of custody every time a bit of critical information is touched as shown in figure 3.

2. Automated access tracking through smart contract will help in detection when copies of evidence are being made and record them in the blockchain, but authorized copies or other types of housekeeping or record keeping activities would be specially entered into the chain of custody of blockchain.

A. Forensic-Chain in Action

Forensic-Chain is initiated or triggered by First Responder,taking hash of digital evidence and recording them securely on blockchain through smart contract. Other details like location, time, and date etc. of crime scene also gets recorded on blockchain. During the course of digital forensics investi-gation any evidence transfer gets automatically recorded on the blockchain through smart contract, recording details like address to whom evidence is transferred to, current state of evidence, permission level, date and time etc. Further any subsequent access to digital evidence also gets recorded securely on the blockchain by smart contracts triggered by corresponding rensic



investigator. Consequently, a chain of trust gets established by recording every action about digital

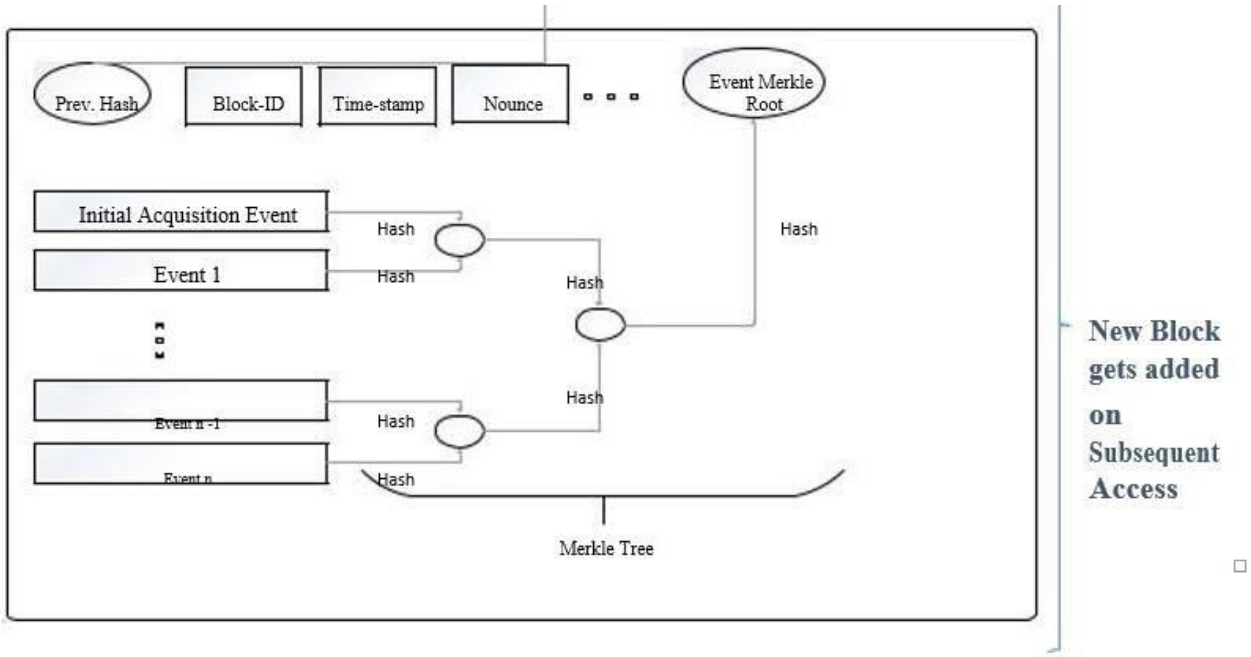


Fig. 3. Forensic-Chain on Subsequent Accesses

evidence, since it originally entered the process in question as shown in figure 4.

B. Benefits of Proposed Model

Forensic-Chain: Blockchain based digital forensics chain of custody has great potential to bring substantial benefits to forensic applications, by maintaining integrity, transparency

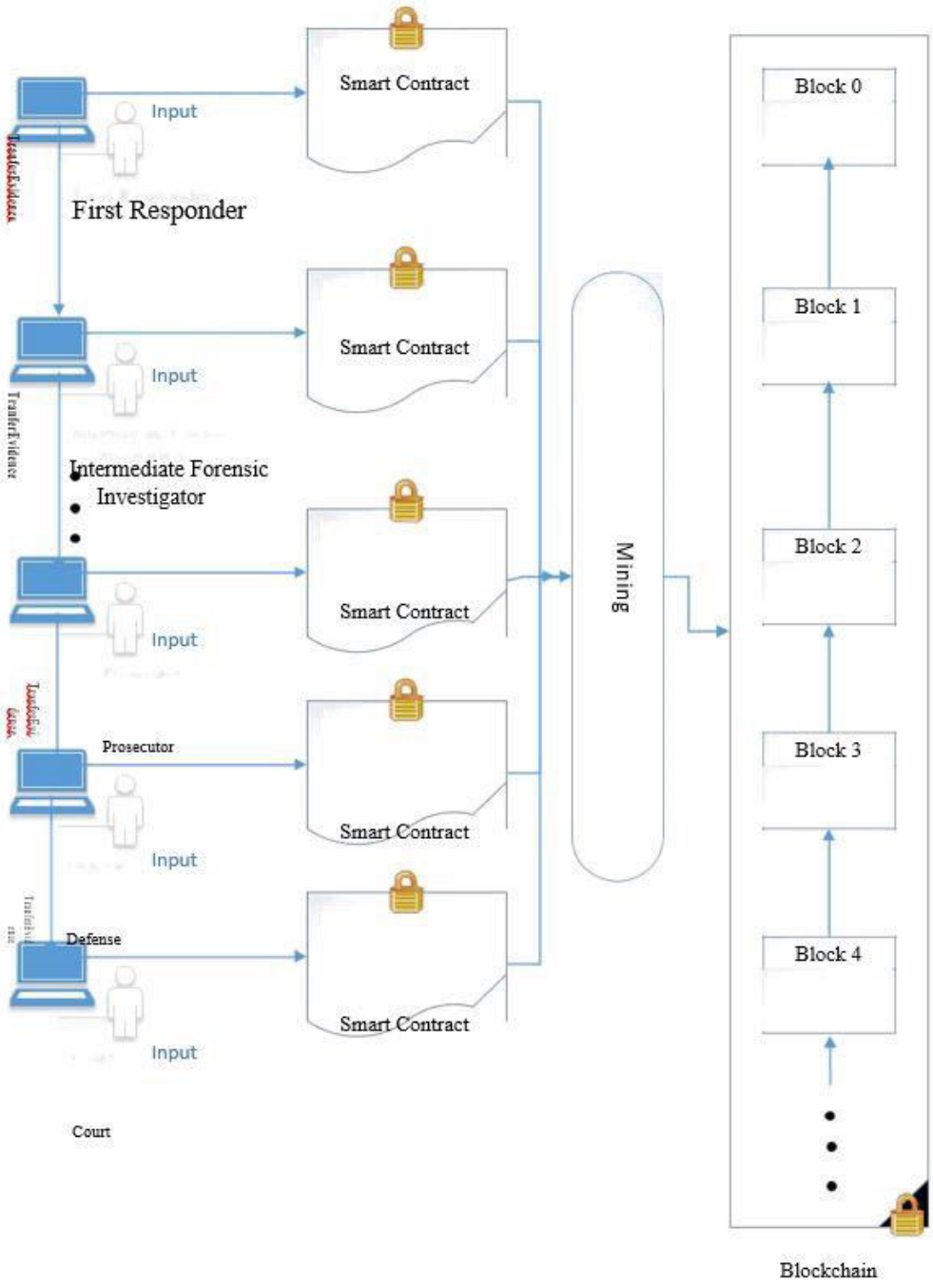


Fig. 4. Forensic-Chain in Action

authenticity, security and auditability of digital evidence to achieve the desired end. Some of the benefits are summarized below:

- Collecting, preserving and validating evidence can be strengthened with the help of Forensic-Chain.
- The provenance of any event or action can be traced back to where it originally entered the process in question.
- Forensic-Chain also helps in improving transactional efficiency and cost reduction of certain kind of transac-tions due to increased transparency resulting eliminat-ing the requirement of trusted third party for validation of certain claims or evidence transfer and consensus based Proof of Trust, resulting increased trust among communicating parties.
- Reduction of fraud due to increased transparency of the audit trail.
- Forensic-Chain allows organizations to embed veri-fication for the event or action within the evidence
- Record itself, thereby enabling an established and ongoing evidence which is both accessible verifiable.

CONCLUSION

Blockchain by design enforces integrity, transparency, au-thenticity, security and auditability thus making it possibly the best choice for maintaining and tracing forensic chain of custody. Blockchain helps in friction reduction through increased trust and thus brings the real promise for foren-sic community. The future work aims at developing complete Ethereum based smart digital forensic chain of custody using smart contracts.

REFERENCES

1. G. Giova, “Improving chain of custody in forensic investigation of electronic digital systems,” International Journal of Computer Science and Network Security, vol. 11, no. 1, pp. 1–9, 2011.

2. S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008.

3. V. Buterin et al., “Ethereum white paper,” 2013.

4. C. Liu, “How the blockchain could transform the process of documenting electronic chain of custody.” [Online]. Available: https://venturaerm.com/Blog/9.html

5. K. Zatyko, “Improving cyber forensics cybersecurity through block chain technology with truth based systems,” International Symposium on Forensic Science Error Management, July-23-2015

Scientific and Practical Cyber Security Journal (SPCSJ) 1(2):28-38
Scientific Cyber Security Association (SCSA), 2017 ISSN: 2587-4667

PERSPECTIVE STEGANOGRAPHIC SOLUTIONS AND THEIR APPLICATION

S. Toliupa, A. Romanova

Taras Shevchenko National University of Kyiv, the Faculty of Information Technology, Information Security Management

ABSTRACT

Nowadays, as the role of information technologies in our lives is growing and growing with every passing minute, there is no question about the fact that information security measures are an issue of the highest importance. It would not be wrong to say that there is no 'zero-day threat' anymore; it is more like 'a threat of a zero second'. Every modern technology automatically causes several new vulnerabilities and threats to come to existence, which, on its part, makes security specialists work their best to counter such efforts. Thus, there are quite a lot of information security solutions at hand already.

Cryptography is doubtlessly one of the most effective, developed and approbated methods to be used when it comes to the protection of information resources. Nevertheless, it might be more effective to hide the communication channel itself instead of making unreadable the information within it. The practice of concealing data within text- or media-file is called *steganography* and has its roots deep in the history of the humankind.

A considerable number of steganographic methods are well-known and implemented in various steganosystems and applications. There are some methods of information concealment, though, that receive the attention not so much. The reasons of such lack of popularity can differ depending on specific solutions: their complexity for one, low cost-effectiveness of their realization for another. In any instance, they are either poorly described or are not widely used regardless of their perspectiveness.

The purpose of this article is to conduct an analysis and suggest possible practical use of steganographic solutions that are known and can be applied to a variety of information security systems, and yet lack either theoretical basis or practical application.

KEYWORDS: Stenography, cryptography, applications, security

STEGANOGRAPHY AS A MEANS OF HIDING INFORMATION

Basic terminology

Steganography is an art and science of storing and transferring secret messages within covert channels that are based on and created inside open channels in such a way that the cover data is perceived as if not having any embedded messages for its unplanned recipients. The general approaches are:

- Full concealment of the covert communication channel;
- Creating difficulties for detection, retrieval and modification of hidden messages conveyed within open data;

- Secret message camouflage inside the protocol [1, 2].

The main concepts are:

- Container *b* (also: carrier) is open data used to conceal secret information;
- Message *m* (also: payload) is secret information to be concealed;
- Key *k* is secret information that is known only to a legitimate user and defines a specific concealment algorithm;
- Empty container *c* (also: unmodified container) is a container devoid of any secret data; it is a sequence of l_c-long elements;
- Modified container *s* (also: package, steganogram) is the one that contains a secret message;
- Steganographic algorithm means two transforms, a direct $F: M \times B \times K \rightarrow B$ and an inverse one $F^{-1}: B \times K \rightarrow M$;
- Steganographic system (also: steganosystem) is a totality of messages, secret keys, containers and transforms that connect them [1, 3].

Steganography uses containers of different nature; the choice depends on the specific task. In case a digital sound file is used as a container the length of elements will be defined as the number of counts per time unit. If the container is a digital image file, then the sequence of elements will be obtained by vectorizing the image (transforming the array of pixels into a vector) [3].

Most steganography methods are based on two key principles:

- Human senses cannot distinguish slight changes in colour, shape and sound perception;
- Consequently, there are files that do not demand absolute preciseness and therefore can be modified without losing their functional value.

As a result, said methods imply allocation of insignificant fragments of the container and replacement of the information within them with information that needs to be hidden.

Finally, the process of encoded steganogram detection is called *steganoanalysis*.

The interest in steganographic solutions is undeniable. They may be used as an alternative for the cryptographic means of hiding information, in many cases more effectively. Furthermore, they serve as a powerful tool for digital and non-digital watermarking and authentication control. Finally, while the use of cryptosystems is legally regulated and limited to some point, there is no such restrictions in designing and distributing steganosystems in any country.

CLASSIFICATION OF STEGANOGRAPHIC METHODS

By the method of selecting a container one can distinguish non-alternative, selective and constructive methods of steganography [3].

Non-alternative methods imply the choice of the first possible container made in order to conceal a message. Selective methods imply that a covert message has to reproduce special statistical characteristics of the container noise. In constructive methods a container is generated by the steganosystem itself.

By the way of access to the secret information there are methods for stream and fixed containers.

By the type of organization there are methods for systematic and non-systematic containers. In the first ones bits of noise and of the container itself can be distinguished. In the latter ones they are impossible to specify.

By the concealment principle there are two main classes: methods of direct substitution and spectral methods. The first use container redundancy and replace the insignificant areas of the container with the bits of a secret message, while the second hide the data using the spectral representation of the elements in the environment where the concealed data is embedded (for example, coefficients of the arrays of Fourier transforms).

By purpose one can distinguish the methods for the data secret transmission or storage and methods for concealing data in digital objects in terms of copyright protection.
A special group is represented by methods that use characteristic file format properties:

- Reserved fields, which are usually filled with zeros and are not taken into account by the program;
- Special data formatting (the shift of words, sentences, paragraphs or selecting specific positions of characters);
- Erasing file identifier captions etc [3].

Popular steganographic solutions

In this section the brief overview of widely used steganographic solutions is presented.
Mostly, steganography uses the data concealment within digital images and audio files, less so video files and text. Electronic communications may also include hiding data inside of a transport layer (program or protocol) [4]. Digital media files are extremely suitable for steganography tasks, first and foremost due to their large size. The subtle changes in their structure are highly unlikely to be noticed by the unintended user.
Starting with non-digital methods, physical steganography technics cannot be omitted. They have been developing for centuries and include, for example, blinking one's eyes in Morse code to spell a secret message [5].
Another example is adding tiny yellow dots to each page while printing a document. They are not detectable by the bare eye and contain the model, serial number and timestamps. This information cannot be obtained from a computer file and is embedded in a printout using dot-matrix code. Such a technology is used by many brand color laser printers, such as Xerox and Hewlett-Packard for traceability reasons [6].

Methods of embedding data within an image container [2, 3]:

- Least Significant Bit method (LSB) (Sequential Insertion) is the most popular steganographic method. The least siginficant bit of each pixel is in fact a noise. If it is changed, the difference in the image will not be noticed by a human eye. Thus, these bits can be replaced with the bits of a secret message.
- LSB Psuedo Random Insertion. In contrast to the previous method, in which every changed data bit follows the next, this method uses pseudo random distribution of the secret message bits through the container. Thus, the interval between two bits is pseudo-randomly defined, which complicates both visual and statistical attacks, as well as extraction of all the hidden bits.
- LSB Pseudo Random Permutation. Not only the least significant bits are chosen pseudo randomly, but also the bits of a secret message are uniformly distributed through the container in a pseudo random sequence.
- Block hiding method. The container is split into disjoint blocks; for each of them a parity bit is calculated. One secret bit is concealed within one block. If the parity bit does not equal the respective secret bit, then one of the LSB in the block is inverted, so that the parity and the secret bits are the same.
- Palette permutation. Any colour palette consists of pairs of indexes. Each pixel of the image corresponds to a certain index in the table. The sequence of colours in the palette is not important, so it is possible to conceal a covert message by changing this sequence.
- Image Quantization. Interpixel correlation can be defined by a function Θ. We can calculate the difference ε_I between adjacent pixels c_i and c_{i+1} (or c_{i-1} and c_i) and set it as a function parameter: $\Delta_i = \Theta(c_i - c_{i+1})$, where Δ_I is a discrete approximation of the difference of signals $c_i - c_{i+1}$. As Δ_I is an integer and the difference $c_i - c_{i+1}$ is a real number, quantization errors $\delta_i = \Delta_i - \varepsilon_i$ occur. The information concealment is carried out by correcting the difference signal Δ_i.

- Kutter-Jordan-Bossen method. A human eye is the least sensitive to the blue colour. The method is based on the embedment of the secret message within the blue channel.
- Koch-Zhao (Relative DCT (Discrete Cosine Transform) values change method). Initial image is split into blocks of 8x8 pixels. As the result of applying DCT to every block a table of DCT coefficients is formed. Every secret bit is hidden in a separate block. Frequencies quantization causes some rate of distortion in the image, which is still not noticeable by the human eye.
- Benham-Memon-Yeo-Yeung method. Optimized version of the previous method. Firstly, only the most suitable blocks are used. Secondly, three DCT coefficients are selected instead of two, which decreases destortion in the container.
- Hsu-Wu method is an algorithm of a binary digital watermark embedment. The value of its pixels can only be “0” or “1”, so the direct observation of such an image is impossible, as “0” and ”1” intensities correspond with the black colour. The watermark can be created black-and-white and then the whole array can be divided by 255 to replace the intensity of white pixels with “1”.
- Fridrich method implies a cascade embedment in low- and high-frequency DCT coefficients.
- Spread-Spectrum method consits of three possible variants:
  - The used frequency band is much wider than needed. Signal/noise ratio is then quite low, which makes the signal unlikely detectable;
  - Spectrum is expanded by using a special independed (also: code) signal. The signal energy is distributed through all frequency bands, which makes the signal noise immune;
  - Restoration of the initial information is carried out by comparing the received signal and a synchronized copy of the code signal.
- Embedding pictures within video-files [5].

Audio steganography [3]:

- LSB-method for audio-files is the same as for images, but working with the audio-file format. It causes considerable destortions in the container.
- Phase coding method implies the substitution of the initial sound segment phase with the reference phase, which is the data to be concealed. Phases of adjacent segments are agreed to preserve the difference phase between them.
- Echo-signal use. Data is embedded in the container by injecting an echo-signal in it. Three echo-signal parameters are changed: initial amplitude, attenuation and shear rate. The echo-signal is perceived only as an additional resonance [7].

Linguistic steganography [3]:

- Random interval methods. Changing the number of spaces in the end of the text string does not cause significant changes in the meaning of the sentence. What is more, an average reader is unlikely to detect insignificant space modifications:
  - Changing the interval between sentences. One or two additional spaces are added after the sentence. This method requires that a file of a considerable size is used to embed a small number of secret bits in. In addition, most text editors automatically change redundant punctuation and spaces, which may ruin the concealed data;
  - Changing the number of spaces in the end of text lines. Spaces are added according to the secret bit to be hidden. Two spaces encode one bit a line, four spaces – two bits etcetera. Compared to the previous method, the bigger amount of information can be embedded.
  - Changing the number of spaces between words in a flattened text. When the text is width aligned, spaces between words are not of the same length and some of them can be used to hide data.
- Making the text of the same colour as the background [5];
- Using similarly looking Unicode and ASCII characters [4, 8];

- Using non-printable Unicode characters [8];
- Creating a pattern of deliberate errors and/or marked corrections [4].

Format steganography:

- BMP:

o Appending data to the end of the file implies an artificial expansion of the final image sector;

o Palette permutation.

- JPEG:

o Appending data to the end of the file. Using the standart sysem of markers to append information after them, which will cause a program to ignore the secret message;

o Using collateral data. Data is preliminarily camouflaged as collateral information (Scan Index, Title Index etc), which is mostly ignored by programs, and then injected after specific identifiers.

o Using commentary markers is similar to the previous method, but works with commentary fields.

Some other methods:

- Converting a file so that it has the statistical characteristics of another one [4];
- Injection of delays to packets that are sent over the netwotk from the keyboard [5];
- Blog-steganography. Secret data is added as commentary pin boards on social networks[5].

Finally, there are different software applications that use the methods of steganographic concealment mentioned above:

- Using LSB-method: OutGuess, JSTEG, JPHS, Hide-and-Seek, Steganos, Steghide, DC-Stegano;
- Using the palette permutation: Gifshuffle;
- JPEG format: OutGuess, JSTEG, JPHS;
- GIF format: Gifshuffle, Hide-and-Seek;
- BMP format: Steganos, Steghide;
- PCX format: DC-Stegano;
- LSB-method in audio-files: Invisible secrets, Hide4PGP, Steghide, StegoWav, Steghan, S-Tools;
- Using parity of quantization of frequency coefficients: MP3Stego;
- Using incorrect frames in a compressed stream: UnderMP3Cover [9].

Perspective steganographic solutions and their application

Internet of Things and cyber-physical systems

A cyber-physical system is a mechanism that is controlled or monitored by computer-based algorithms, tightly integrated with the Internet and its users. Examples of CPS are autonomous automobile systems, medical monitoring, smart grids, automatic pilot avionics etc [10].

The Internet of Things (IoT) is the network of physical devices, vehicles and other items embedded with electronics, sensors, software and network connectivity, which enable them to collect and exchange data. It is more or less an instance of a class of cyber-physical systems [11]. The network steganography uses communication protocols' control elements and their functionality to hide information inside [12]. The modifications can be carried out either over a single network protocol (applied to the Protocol Data Unit, the time relations between PDUs or both) or to several protocols at the same time (inter-protocol steganography). Such network steganography methods can be applied to the systems mentioned above, too. The IoT is believed to be a phenomenon that will expand its influence greatly within the next few years. As a perspective network instance it requires thorough attention of steganography specialists.

Information circulates within it the same or the fairly similar way as in any other system. Thus, optimal and the most suitable methods of hiding data in communication protocols should be developed specifically for the IoT.
What is more, as the items within the IoT possess a vast variety of sensors and software, they can be used to conceal data in. For example, covert messages can be stored in unused registers of the CPS/IoT components or in the states of their actuators [12].

THE USE OF STREAM CONTAINERS

As mentioned above, by the type of access to the data one can distinguish fixed and stream containers [3]. All the methods mentioned in Chapter 2.3 use the first ones to conceal information in. Such a container is a constant pre-defined sequence of bits that are displayed before a steganographer all at once. To the contrary, a *stream container* is a sequence of bits that are continuously changing, as in a phone conversation. A message is embedded in real time so that the final size of the container is never known beforehand. The intervals between the embedded bits are generated by a pseudorandom sequence (PRS) generator and uniformly distributed between readouts [3].
There is hardly a couple of scientific works devoted to this type of steganography, let alone examples of its real-life practical implementation. Despite any reasons, it can be successfully applied as an efficient means of information security. There is a number of solutions for encrypted secure real-time communication. However, what if we could, for example, make a confidential phone conversation not only indecipherable but also seem to be an innocent chat? A stenographic telephone set-top box could be a solution. The same concerns video-conferences. An extraneous observer would only see an average conversation not having any access to the real audio, video or any other embedded data.
The unpopularity of the stream-container steganography can be explained by defining major issues concerning its use. First and foremost, it is never known whether the size of the container will be enough to conceal the whole message as the length of the first (and likely of the latter, as well) is undefined. The same property creates and advantage as one carrier file can be capacious enough to contain several messages. In any case, the secret data has to be somehow synchronized with the container, thus one of the biggest questions is how to define the beginning and the end of the embedded sequence within the container. The problem becomes more serious concerning video communication. The solution would be of extreme complexity, as we would need to synchronize the image-image stream (both open and covert), the sound-sound stream and image and sound respectively.
The solution may lie in using special built-in libraries. They would consist of structured groups of words of the same length, which would in ideal case possess pronunciation similarities. Such groups should then be grouped in semantic dictionaries, so that they would form simple, but logically and semantically structured sentences. The linguistic means for this are well-developed and are similar to those of forming synonymic dictionaries and machine translation applications. The words and sentences could then be synchronized with the container using synchronization bits, package headers and/or other means of dividing encapsulated data; the covert message can be embedded after them and be synchronized using the initial properties of the container.
The possible situations with video communication would be more complex. If only the content of a given conversation is confidential, then the issue is just to steganographically encrypt the sound and synchronize it with the real video image. On the other hand, if the identities of conversation participants are also a secret, then other methods should be provided. It is not necessary for a steganographic solution to be all-purpose. It is possible to design a system consisting of a cryptographic and a steganographic modules and providing different scenarios according to the situation.

The biggest remaining problem is a significant delay which is unacceptable in real-time conversations. Then again, there are numerous solutions in cryptography in this field, that can be adapted to the task.

SEMANTIC AND SYNTACTIC METHODS

These two classes of methods belong to steganography with text containers. Instead of using digital format features, they work with the language itself. It is an advantage in comparison to the first type. As an average reader may not be aware of the covert message existing in an open text, a text editor may automatically change the number of spaces or conduct other actions that would ruin embedded data [3]. In fact, any reformatting will lead to the same result.

Syntactic and semantic methods, though, do not use the presentation of text, but work with the text itself. The first type uses the fact that in most languages there are some optional rules of punctuation and grammar forms. Any given language sticks to specific rules, but is still not so solid of a structure, which presents a great number of linguistic possibilities. For example, in the Ukrainian language a colon and a dash can replace each other in some cases. This can be used to encode bit of secret message: “0” for one punctuation mark and “1” for the other one. A more complicated method could be using grammatical similarities in different sentence constructions, such as changing the sequence of some words.

An example of semantic steganography is using the table of synonyms to encode the secret bits. If there are two of them, say, ‘however’ and ‘but’, then again one of them can mean “0” and the other “1”. If there are more synonyms, possibly context ones, then 4 words can encode 2 bits, 6 words 3 bits of information etcetera. The average data transfer speed when using these methods is several bits per kilobyte [3].

The main problems with such linguistic methods are obvious. First of all, they are very language-dependent. Secondly, they require large amounts of initial text as a container, which is not exactly effective. Finally, even if some punctuation rules are ambiguous, their deliberate and controversial usage can be detected by a censor/editor.

It could be wise to suggest the usage of more complex methods of language-based steganography. Every language can be analyzed to create special tables of syntactic correspondence. For example, for the English language and other Germanic languages the use of active and passive voice, as well as of complex object and complex subject is optional. Sentences can be easily and naturally transformed using equivalent constructions, that most likely will not raise suspicion. The advantages of such solution are numerous. One of them is high resistance to various attacks (they are here similar to one-time pads). Another is that the concealment capacity is much higher than that of basic semantic methods. The only question is an algorithm of selecting initial text material. It is likely the best option is to create special libraries of texts, sorted by the topic. This way many fields of interest may be covered so that the covert message is not detected.

A creation of a multi-purpose linguistic steganography complex is suggested. It will doubtlessly require linguistic work of high quality and profoundness. An optimal approach is to be found to, if possible, reduce the language-dependency of each solution. In other case, such an application will have to be designed according to a separate language or, at least, a group of languages with the same paradigm. Thus, the task at hand is to group the languages within each family by the similar tendencies in grammar usage. The next step is to create tables of correspondence for grammatical constructions and stylistic expressions that can be interchanged. Finally, text material libraries are required to provide unobtrusive containers with as much options described in the tables as possible.

STEGANOANALYSIS

Methodological base of steganographic analysis also require some further enhancement:

1) Development of probabilistic-statistical methods of recognition, application of artificial intelligence elements to estimate the reliability of steganographic transforms and to design detectors (filters) for analyzing information streams in order to detect and overturn hidden communication channels. In this case, the verification of the hidden information presence is specified by a certain estimate using statistical criteria (sequential correlation, entropy of the image, dispersion of the LSB, etc.). Solutions developed for this purpose should not only provide a low rate of error in the recognition of incoming messages (especially when using encryption of the data), but also be able to detect messages embedded using different steganographic methods. Compared to the applications designed to steganographically conceal data, the quantity and quality of steganoanalytical systems are rather low.

2) Analysis of specific steganographic software solutions to restore the concealment algorithms and work out the optimal analysis method. The main difficulty is again the large number of specific algorithms that demand individual approach as well as significant amount of computations.

3) Development of the technology of automized active and malicious attacks to make the anticipated steganogram irreducibly distorted in order to provoke its re-transmission within another container, which would confirm the existence of a covert channel [3].

BIOCHEMICAL STEGANOGRAPHY

Most modern steganographic systems use only digital containers, such as files of various nature, binary sequences etc [9]. However, there are other fields of interest for steganography, as the environment can provide a considerable variety of non-digital containers.

We are surrounded by billions of organisms, every cell of which contain DNA-molecules. They are the central repository of information in the cell [13]. Biological computing and quantum computing are believed to be the two most promising technologies under development right now [14]. And as cryptography now mostly works with factorization problems, which makes the messages subjects to attack by quantum computers, biochemistry presents the whole new sphere of potential information security solutions.

DNA-steganography is a process of camouflaging a DNA-encoded message within the enourmous complexity of genomic DNAs [13]. Due to the DNA-code variability among different species, an organism, selected at hazard, possesses random DNA-code. Such a characteristic makes these molecules potentially good containers. Another doubtless advantage is their tiny size, as huge amounts of information can be encoded within a container that cannot even be seen by a human eye without proper amplification. DNA is also a quite solid structure and one highly resistant to possible biochemical attacks.

Nevertheless, despite obvious perspectives of using DNA as a means of biochemical steganography, most of the attention has been received by DNA-cryptography so far.

A DNA-molecule is a sequence of four nucleotides – Adenine (A), Cytosine (C), Guanine (G) and Thymine (T), that are grouped in triplexes, so called *codones*, and form two anti-parallel strands [9, 14]. Complementary DNA strands can self-assemble by forming hydrogen bonds between bases (base pairing) of each strand specifically with A bonding only to T and G only bonding to C [10]. Four different bases mean 4^n possible different n-meres that encode genetic information through a number of aminoacids [6, 8]. Another macromolecule to be potentially used is RNA. It is similar to a DNA-molecule with an exception that the base structure is a different 5-atom sugar – ribose, and uracil (U) corresponds to thymine [9].

So, how do we encode information within a biomolecular structure? A message can be encrypted in a DNA strand, every symbol being encoded by a codon defined in the specially designed table (the technic resembles the use of one-time-pads). For example, ‘A’ may be encoded by a CGA

sequence, ‘B’ by CCA and so on [14]. The secret message is then presented as a sequence of codones. Some of the aminoacids are presented: alanine – GCT, GCC, GCA, GCG; asparagine – GAT, GAC; fenilalanine – TTT, TTC [9].

Then the strand is flanked by polymerase chain reaction (PCR) primer sequences and hidden by mixing it within many other additional “distracter” DNA strands [10, 9]. Polymerase Chain Reaction (PCR) is a process, during which PCR primers become complimentary to the F and R primer “keys” in Secret Message DNA [13]. They are then hidden in a microdot [14]. Knowing the secret key and the primer sequences, a user can extract the strand using known DNA separation methods (hybridization with the complements of the “secret key” strands might be placed in solid support on magnetic beads or on a prepared surface; may be combined with amplification steps and/or PCR [15]) and read the message.

A problem with such approach is that the probability profile of aminoacids in nature is not the same as that of a secret message [9]. The secret “tags” have to be indistinguishable from “distracter” DNA strands and the entropy has to be as in any DNA-molecule – between 1.2 and 2 [15]. This creates the need to use models of real DNA-molecules along with some other solutions. One of the enhancement technics suggested recently is the use of *sequencing* (determining the sequence of nucleotides in a DNA-fragment). There are a lot of sequenced genomes provided in open arrays already. Some of them are 55 genomes of bacteries, a yeast genome and those of other standart laboratory objects [6]. Another techinc is to construct the “distracter” strands so that their distribution mimics the plaintext source distribution. One of the easiest possible ways to do so is to synthesize a DNA-molecule that depends only on the plaintext and the secret key [9, 15]. The compression of the plaintext is also possible. If the resulting distribution of the plaintext approximates a universal distribution, then a random distracter sequence may suffice to provide security needed [15]. The use of a substitute random combination of sequenced genomes (from exotic organisms, for example) may be an enhancement solution, as well [13]

There is a work [16] devoted to the use of run-length encoding (RLE) systems in biochemical steganography, though it is stated, that their application in practice still provides more questiones than answers.

Given everything stated above, any possible attack on a DNA-based steganographihc system would not be successful if it is purely computational [13]. The ways of resisting biochemical attacks, though, are an important question to pay attention to in the future development of DNA-steganography.

Current DNA-steganography technology is still in a period of laboratory exploration and focused on experiments [14]. A possible explanation of the lack of expected activity in the field is that it is a multidisciplinary area which demands knowledge in both biology and cryptography and so requires researchers from both areas to work in a new cooperative way [14]. Possible spheres to implement these technics in are negotiable instrument anti-forgery, personnel identity and access control, anti-theft marking and product authentication [17]. All of them are instruments of securing business profit and are thus attractive for service and production. Only a few examples of using DNA-steganography as a sort of watermark are know. In 2000, during the Olympics in Sydney the Australian Olympic Committee used the DNA based tracking technology to protect Sydney Olympic licensed merchandise from counterfeiting [17].

CONCLUSION

A variety of steganographic solutions was analyzed. Among them such methods were selected that are not either widely used in software applications or lack attention in general. Nevertheless, their perspective usage was discussed. Taking into account all the fact mentioned above, the next directions of development of steganography are suggested:

- Steganography in cyber-physical systems and the Internet of Things in particular;
- The use of stream containers;
- Semantic and syntactic methods;
- Some enhancement in steganoanalysis technics;
- Biochemical steganography practical application.

Information is surely becoming an asset of the highest value. Seeing as the cyberspace is more of a battlefield for different forces continuously confronting each other, it is obvious that information security sphere requires the best solutions possible. Steganography has proven to be an effective means of secret data concealment ensured with centuries of practical use. And, just as any other science, it is in the state of constant development. Being aware of perspective ways to use its methods for our cause, we get access to numerous up-to-date possibilities of providing information security of the highest level.

REFERENCES

1. Е.Л. ЗОРИН, Н.В. ЧИЧВАРИН: Стеганография в САПР. Учебное пособие. МГТУ им. Н.Э. Баумана, Москва (pdf).
2. ALEXANDRE MIGUEL FERREIRA: An Overview on Hiding and Detecting Stego-data in Video Streams. University of Amsterdam, System & Network Engineering – Research Project II, March 23 2015.
3. KONAKHOVICH G. F., PUZYRENKO A. YU.: Computer steganography. Theory and practice with Mathcad (Rus). MK-Press Kyiv, Ukraine 2006.
4. FRIDRICH, JESSICA, M. GOLJAN, D. SOUKAL: Searching for the Stego Key. Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VI 2004 (pdf): http://www.ws.binghamton.edu/fridrich/Research/Keysearch_SPIE.pdf.
5. CHRISTOPHER LEAGUE: An overview of digital steganography, particularly within images, for the computationally curious. Long Island University 2015: https://www.youtube.com/watch?v=-7FBPgQDX5o.
6. Secret Code in Color Printers Lets Government Track You; Tiny Dots Show Where and When You Made Your Print. Electronic Frontier Foundation October 2005: https://www.eff.org/press/archives/2005/10/16.
7. Echo Data Hiding (html): http://www.slidefinder.net/a/audio_steganography_echo_data_hiding/ 24367218
8. AKBAS E. ALI: A New Text Steganography Method by Using Non-Printing Unicode Characters. Eng& Tech. Journal, 28 (1) 2010 (pdf): http://www.uotechnology.edu.iq/tec_magaz/volume282010/No.1.2010-/researches/Text%287%29.pdf.
9. А.В. АГРАНОВСКИЙ, А.В. БАЛАКИН, В.Г. ГРИБУНИН, С.А. САПОЖНИКОВ: Стеганография, цифровые водяные знаки и стеганоанализ. Москва: Вузовская книга 2009.
10. Cyber-Physical system: https://en.wikipedia.org/wiki/Cyber-physical_system.
11. Internet of Things: https://en.wikipedia.org/wiki/Internet_of_things.
12. STEVEN J. MURDOCH, STEPHEN LEWIS: Embedding Covert Channels into TCP/IP. University of Cambridge, Computer Laboratory (pdf): http://www.cl.cam.ac.uk/users/fsjm217, srl32g/.
13. CARTER BANCROFT, PH.D.: DNA-Based Technologies: Computation, Steganography, Nanotechnology. Talk at Material Science and Engineering, Stony Brook University, April 2011.
14. ADITIT SHARMA: Security and Information Hiding based on DNA Steganography. International Journal of Computer Science and Mobile Computing, Vol. 5, March 2016: www.ijcsmc.com.

15. ASHISH GEHANI, THOMAS H. LABEAN, JOHN H. REIF: DNA-based Cryptography. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Volume 54, 2000 (pdf).
16. TOMONORI KAWANO: Run-length encoding graphic rules, biochemically editable designs and steganographical numeric data embedment for DNA-based cryptographical coding system. Commun Inteqr Biol. March 2013: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3609851/
17. WENDELL M. SMITH: DNA Steganography for Security Marking. 5th World Product and Image Security Convention, PISEC '03, Czech Republic. Technology Transfer Group: http://www.polestarltd.com/ttg/isspeeches/pisec03/index.html

Scientific and Practical Cyber Security Journal (SPCSJ) 1(2):39-47
Scientific Cyber Security Association (SCSA), 2017 ISSN: 2587-4667

THE PURPOSE AND TASK OF BUILDING COMPLEX DISTRIBUTED SYSTEMS, THE PARAMETERS OF STABILITY AND SECURITY IN THE FORMATION OF THE SYSTEM DEVELOPMENT GOALS OF ITCS

Y. Shestak, V. Vialkova, S. Mahula L.Mirutenko
Taras Shevchenko National University of Kyiv

ABSTRACT:
The main purpose of the establishment and operation of distributed information and telecommunication systems is the organization of effective user access to information and software resources, and effective interaction as users with the resources and various kinds of resources between themselves. There is also the importance of considering the stability parameter distributed ITCS in the context of its functional stability. Admittedly, distributed systems can potentially have a higher resistance to failure, compared to centralized structures. There also several objectives of sustainability and security of distributed ITCS The network's security settings, sustainability and security of the system against internal and external effects are significant. That is because of ensuring the implementation of the key requirements of availability, confidentiality and integrity of data, processes and procedures inside the system.

KEYWORDS: distributed information and telecommunication systems, information security, stability

Distributed systems are created in order to help users to access remote resources. The notion of resource here must be understood in a broad sense: from devices (printers, scanners) and ending with files, web pages and the like. Sharing of resources is justified in the first place economically. The use of shared resources also enables the rapid exchange of information between users. At the same time, easy access to resources should not be a resignation for the deterioration of stability and security distributed ITCS, reducing security operations performed in the system. It concerns issues of both privacy and integrity of transmitted data, a reliability of

the system.

As complex systems, distributed ITCS represent a set of interrelated technical and technological elements, functioning under the action of random factors, in an active interaction with the external environment, in the presence of negative influences of different nature and high cost of the consequences of possible violations or errors in the system [1].

Describing the basic concepts concerning security issues of distributed ITCS should define "fails" – the state when the system does not perform its functions; is not acting under a specific protocol [2,3].

Eror – the part of the system state, which can reach up to the accident (eg. error data broadcasts).

A vulnerability is the cause of the error.

Vulnerabilities control is performed through prevention, removal and expectations.

Failure – the visible inability to perform the function of the system, which is a consequence of errors due to vulnerabilities.

Fault tolerance– the ability to provide services despite the existence of vulnerabilities..

In the formation of development goals distributed ITCS significant place should be occupied by the parameters of the stability and security of complex distributed ITCS, which will allow using certain methods, information technologies and tools to ensure the stable operation and development (using the properties of scaling) distributed ITCS under the influence of these threats [4;5] (table 1.2):

Table 1.2 The main threats types to the sustainability and security of distributed ITCS

Threat types	Essesance	The motives and methods of cyber criminals
Passive attacks	Provide motion analysis, monitoring unsecured communications, decoding network traffic that is encrypted with weak cryptographic algorithms and intercepting information	Passive reception of information on the network can serve as a source of information for cybercriminals, providing certain data and possibilities of interference with the system without the need for user consent.
Active attack	Involve attempts to bypass or hacking, using, for example, special software, or theft or	These attacks may be to use the transmitted information, the attacks on the authenticated user

	modification of information	when you attempt remote login to the system. Active attacks can serve as identification or dissemination of data, blocking delivery of services
Convergence	One is obtaining physical access to network, access or block access to information	Convergence can be accomplished through covert or overt actions can also be the result of a combination of these two methods of action
Attack from within:		It should be remembered that more than 80% of attacks on systems is carried out from the internal environment of information and telecommunication networks [6]. Given this, users on the network should not have more opportunities than they need and all their actions should be recorded for the possibility of establishing the source of danger
- M*alicious attacks*	One is obtaining physical access to network, access or block access to information	
- *Not malicious attacks*	As a rule, impose a frivolity, a lack of competence	
Distribution	It is the adaptation of equipment and software at the time of manufacture or distribution	Attacks of this type are the inclusion in the product, which is used in distributed ITCS code, the so-called back door, thanks to which cyber criminals will continue to have access to systems that use these products

The growing complexity of the structure and functioning of complex distributed ITCS leads to the emergence of such properties as a natural redundancy, adaptability, reliability, fault tolerance, resilience, survivability [6].

Functional stability and security of distributed ITCS characterizes its ability to implement fully their roles and perform tasks, for which the organizers distributed ITCS receive income from providing services [7,8].

The stability of the system describes its ability to operate in a continuous mode. Unlike availability, description of sustainability refers to the period of time. It is the quality that allows ITCS distributed smoothly to withstand changes in the parameters of the external environment different from the design and carry out their basic functions. Thus, distributed ITCS can be considered sustainable if it can cope with variations (sometimes unpredictable) in the operating environment with minimal: loss, configuration changes or loss of functionality.

This implies the importance of considering the stability parameter distributed ITCS in the context of its functional stability (ability to ensure the functionality of the system under the influence of environmental parameters and their changes). Among the components of the functional stability of distributed ITCS, following ones should be highlighted:

- ensuring of the availability, characterizing the condition in which is stored the user access to the services of ITCS distributed in full without additional (unspecified agreement) access restrictions in time and space;
- ensuring of the integrity of information that characterizes the state, which retains the ability of users to access services distributed ITCS in full and no additional (unspecified agreement) access restrictions in volume), due to the impact of ITCS on distributed.

Security distributed ITCS is a complex characteristic that is characterized by a complex interaction of means and technological methods, as well as procedural, logical, and physical measures aimed at [9,10]:

– countering threats to information resources and components of the information environment;

– protection components of the information environment;

– minimize risk to the components and resources of the information environment.

Thus, the parameters of sustainability and security of distributed ITCS, although they have some functional commonality, differ primarily the orientation to specific tasks, in particular, to ensure sustainability it is important to ensure the uninterrupted operation of the system, for security – combating external interference and threats, which ultimately determines the possibility of stable (without interruptions caused by external interference) of the system.

Distributed systems work by using separate computers connected to the network. From the programmer's point of view, this is important, because the operation of individual machines in a distributed system affects the way of programming such systems. An important parameter in

determining the nature of ITCS distributed architecture connections between nodes, which can be implemented via a central bus or dial-up technology [11].

These systems have a specific purpose of functioning, a large number of interacting subsystems, a complex hierarchical control system. They are also characterized by complexity and functions that they are, the constant rise in the number of users and connected equipment, a constant modification of the components, which leads to the impossibility of constructing an adequate mathematical model for a comprehensive description of the functioning of the system [12].

Most often, when designing a distributed information-telecommunication system in the first place put a division of its functions between multiple computers [13]. This approach is distributed in any computer system where data processing is divided between two or more computers.

Organization of work with information resources in distributed ITCS provides a solution to the complex task of ensuring convenient and quick access to information for those who are entitled to it and protect information from those who do not have appropriate access rights. This leads to the need to solve additional problems of sustainability and security of distributed ITCS related [14]:

- authentication of remote users and programs;
- protection of communication channels;
- protection of remote nodes of the system;
- protection of the entire distributed system as a whole, its management.

These very requirements for durability and security when building distributed ITCS determine today the biggest r problems of a technical and technological plan, the reason for this is that the creation of systems of information security seriously lagging behind in the development of technologies of transfer and processing of information, it is rather a consequence of the reaction to potential threats, rather than a systematic process of preventing instability and insecurity of the system, carried out constantly, at the stage of design and planning of building systems. Distributed systems can potentially have a higher resistance to failure, compared to centralized structures. Partial failure in a distributed system may cause damage to one component of the system that can be replaced by another, and does not require stopping the functioning of the entire distributed ITCS to recover her health. In centralized systems, the failure usually causes partial immobilization of the entire system.

Most fault tolerance of distributed systems describes only its potential, but not necessarily the fact of its stability and security, and requires the use of various supplementary methods of

protection, the implementation of corrective actions in case of failure or unauthorized interference in its activities.

An important feature of many applications is their indivisibility. This applies, for example, to Checkout, where the individual steps must be completed in full. Implementation integrity of a distributed system requires a distributed statement, which is equivalent to consensus.

Ensuring of sustainability and security of distributed ITCS requires the use of a certain number of specific means of information protection. These tools combine hardware and software components into a coherent, complex system of protection. However, there is a certain probability of violation of the stability and security of distributed ITCS, which can be described using the following model:

$$p_i = \frac{NA_{refl.i}}{NA_{tot.\ i}},$$

where – p_i – the probability of loss of stability and security distributed ITCS as a result of the threats of *i*-type;

- $NA_{refl.i}$ – the number of reflection attacks (threats) sustainability and security of distributed ITCS of *i*-type;
- $NA_{tot.\ i}$ – the total number of attacks (threats) sustainability and security of distributed ITCS of *i*-type

The probability of cracking (system security breach which leads to the impossibility of performance of its functions) distributed ITCS as a whole can be modelled in the following way:

$$p_{prot} = 1 - p_1 * p_2 * p_3 * p_4 * p_5 * \ldots * p_n,$$

where p_{prot} — probability of breaking (system security breach which leads to the impossibility of performance of its functions) distributed ITCS as a whole;

$p_{1\ldots}p_n$ – the probability of security breach distributed ITCS as a result of the threats of the *i*-type (i=1…n).

In this model, however, does not take into account the activity of the organizers distributed ITCS in the protection from threats to the sustainability and security of distributed ITCS. Based on the foregoing, a violation of resource availability can be described as:

$$p_{avail} = 1 - (1 - p_1) * (1 - p_2) * (1 - p_3) * (1 - p_4) * (1 - p_5) * \ldots * (1 - p_n),$$

where p_{avail} — the probability of violation of availability (the system loss of stability and

security, which leads to the impossibility of the availability of users to distributed services ITCS) in general.

Violation of the integrity of the information can be presented in the form of the model:

$$p_{int.} = p_j * [1 - (1 - p_{conf}) * (1 - p_{net}) * (1 - p_{other}),$$

where $p_{int.}$– the probability of violation of integrity of information;

p_j – the probability of violation of control and you need to recover information;

p_{conf} – probability of breach of confidentiality;

p_{net} – the probability of a negative impact on information from the vulnerability exploits a telecommunications network;

p_{other} – the probability of a negative impact on information from exploit out of a telecommunication network.

Based on previous dependencies, complex value of the probability of loss of stability and security distributed ITCS can be shown by the following equation:

$$p_{\text{ст.та зах.}} = 1 - (1 - p_{\text{зах}}) * (1 - p_{avail}) * (1 - p_{int.}),$$

where p_{avail} — the probability of system loss of stability and security, which leads to the impossibility of the availability of users to distributed services ITCS;

p_{prot} — the probability of cracking (system security breach which leads to the impossibility of performance of its functions) distributed ITCS as a whole;

p_{avail} — the probability of violation of availability (the system loss of stability and security, which leads to the impossibility of the availability of users to distributed services ITCS) in general.

$p_{int.}$– the probability of violation of the integrity of the information due to the impact of ITCS on distributed.

The security vulnerabilities distributed ITCS and, thus, their causes cause may be very different. Depending on their nature, is necessary to implement relevant activities. Important in this context is the classification of different types of vulnerabilities of security systems.

– Temporary vulnerability – appear and fade, are transient. These types of vulnerabilities may occur through temporary weather conditions, or as a result of transient interference from external devices or animals.

– Fitful of vulnerability also appear and disappear, but their condition is characterized by relapses. An example of this type of vulnerabilities can be, for example, poor contact of

conductors in the network. This type of vulnerability is relatively difficult to detect because they can appear when the monitoring system is not configured to reveal them.

– Permanent vulnerability does not disappear until they are fixed. This is the result of, for example, fault or error in the software.

To describe the maximum effect of neutralizing the threats to the sustainability and security of distributed ITCS i-type (i=1…n) using the tools on your system (d) using the following equation:

$$\sum_{j=1}^{m}\sum_{i=1}^{n} d(i,j)p(i,j) \Rightarrow \max$$

Subject to the restrictions on the costs (C) to ensure the stability and security of the system equations will have the following form:

$$\sum_{i=1}^{n} c(i) * sign \sum_{uj \in U}^{n} p(i,j) \le C$$

$$p(i,j) \in (1,0), \quad j = 1, \ldots m; \quad i = 1,\ldots n.$$

The objectives of sustainability and security of distributed ITCS are:

– the elimination of these vulnerabilities even before the negative impact of events related to vulnerabilities in the system, on the functioning of the system;

– prevention of negative influence of events related to vulnerabilities in the system;

– fixing effect on distributed ITCS of events related to vulnerabilities in the system;

– elimination of consequences of the negative influence of events related to vulnerabilities in the system;

– stabilization of the system operation after system failures caused by events related to vulnerabilities in this system.

The implementation of these tasks is in the process of management and administration of complex distributed ITCS as one of the key elements of the integrated functions and ensure the realization of the main goal of the functioning of modern complex distributed ITCS is the efficient user access to information and software resources as well as the effective interaction of users with resources and different types of resources between them. In fulfilling of this objective, the network's security settings, sustainability and security of the system against internal and external effects are important, because in this way ensure the implementation of the key

requirements of availability, confidentiality and integrity of data, processes and procedures within the system.

REFERENCES:

1. Dodonov A. G., Kuznetsova M. G., E. S. garbacik Survivability and reliability of complex systems. Methodical manual. — International scientific-training center of UNESCO/IIP of information technologies and systems. — 2001. — 163 c.
2. Tsymbal, A. A. the Technology of creation of distributed systems. For professionals /A. A. Tsymbal, M. L. Ensina - SPb.: Peter, 2003. - 576
3. Pleskach, V. A. Informatin technology the system /V. A. Pleskach, V. Rogushina, N. P. Kustova; Kiski NAT. torgovelnye-Econom. UN-t. - K:. “The book”, 2004 - 519.
4. The usage of mechanisms to improve survivability to ensure the security of the information resource in distributed systems / N.G. Kuznetsova // Registration, storage and processing. data. — 2006. — Vol. 8, No. 3. — S. 40-47.
5. Schneier. Network control and security // Protection of information. Konfident. — 2004. — No. 4. — S. 75-81Radchenko G. I., a Distributed computing system / G. I. Radchenko. – Chelyabinsk: Photographer, 2012. – 184
6. Radchenko G. I., a Distributed computing system / G. I. Radchenko. – Chelyabinsk: Photographer, 2012. – 184
7. Domarev V.V. Protection of information and security of computer systems. — K.: Publishing house "Diasoft", 1999. 480 p.
8. Kuznetsova M. G. the Security of information resources in distributed systems: Sat. scientific. Tr. "Information technologies and security". Vol. 7. — K. AND NASU, 2004. — Pp. 38-40.
9. Domarev V.V. Protection of information and security of computer systems. — K.: Publishing house "Diasoft", 1999. 480 c .
10. Kuznetsova M. G. the Security of information resources in distributed systems: Sat. scientific. Tr. "Information technologies and security". Vol. 7. — K. AND NASU, 2004. — Pp. 38-40.
11. Papazoglou M.P. Web Services: Principles and Technology / M.P. Papazoglou // Prentice Hall. – 2007. – Vol. 21. – P. 139-145

12. The usage of mechanisms to improve survivability to ensure the security of the information resource in distributed systems / N.G. Kuznetsova // Registration, storage and processing. data. — 2006. — Vol. 8, No. 3. — S. 40-47.
13. The usage of mechanisms to improve survivability to ensure the security of the information resource in distributed systems / N.G. Kuznetsova // Registration, storage and processing. data. — 2006. — Vol. 8, No. 3. — S. 40-47.
14. Robert J. Ellison, David A. Fisher, Richard C. Linger, Howard F. Lipson, Thomas A. Longstaff, Nancy R. Mead. Survivability: Protecting Your Critical Systems. http://www.cert.org/archive/html/ protect-critical-systems.html

Scientific and Practical Cyber Security Journal (SPCSJ) 1(2):48-54
Scientific Cyber Security Association (SCSA), 2017 ISSN: 2587-4667

DESIGN OF MULTI-USER SYSTEMS BASED ON HUMAN COMPUTER INTERACTION

M.Iavich, G.Iashvili
Scientific Cyber Security Association (SCSA)

ABSTRACT:

It is necessary to take into account special factors and methods of interaction between a human and a computer in order to develop a competent and user-friendly multi-user systems. One of the key points is the security of the system and it is necessary to find optimally convenient ways for users to work with it. Only taking into account main aspects and examples of using these methods in practice, it is possible to create a really handy tool for a wide audience. Another important fact is usability, which is the basis for system development. And developing of the system can only be based on practical experience and user preferences. It should also be noted the importance of regular evaluation and analyzing of complete product. Work on improving and optimizing the system should be done systematically.

KEYWORDS: HCI, usability, security, users, systems.

Для создания безопасной и удобной для пользователей системы необходимо учитывать, как технические характеристики той или иной технологии, так и ключевые факторы способностей человека, и его взаимодействия с новейшими технологиями. В данной статье мы рассмотрим основные критерии оценки HCI - взаимодействия человека с компьютером, их использование на практике, а также их внедрение в разрабатываемую систему на основе полученных данных. Мы рассмотрим примеры использования различных методов взаимодействия пользователей с системами на практике.

Для успешной планировки процесса создания многопользовательской системы за основу берется HCI, и с учетом основных аспектов данного механизма разрабатывается, как и визуальная, так и технически-программная часть системы.

Взаимодействие человека с компьютером HCI (*англ.* Human Computer Interaction) - это изучение того, как человек взаимодействует с технологией. Пользователи могут работать как со стационарных компьютеров, так и через мобильные устройства. Некоторые пользователи могут взаимодействовать с технологиями посредством носимой электроники, на пример с помощью умных часов, браслетов и.т.д.

С точки зрения человека необходимо установить психологические и когнитивные способности пользователя, что дает возможность разработать систему используя возможности рядового пользователя, не заставляя его совершать слишком сложные действия для достижения результата.

С точки зрения технологии, основываясь на HCI можно спроектировать систему учитывая потребности и возможности пользователя, а также его взаимодействие с технологиями. Необходимо так же в процессе работы оценивать пройденные этапы для проверки правильности наших действий. Основная цель заключается в том, чтобы люди не затрачивали больше усилий, чем требуется для использования разработанной технологии.

Начинается все с понимания задач того или иного пользователя в системе. Пользователи могут быть кем угодно, начиная с ребенка, заканчивая уже зрелым человеком, некоторые могут работать группами - другие в одиночку. Задачи, которые выполняют пользователи могу варьироваться от самых элементарных как на пример вход в систему, до комплексных, типа обработки и анализа большого количества информации. И наконец, контекст, пользователи и задачи могут быть одинаковыми, но от того в каких условиях проводятся те, или иные действия зависит работа системы, и влияние на нее различных факторов.

После понимания пользователей, их задач и контекста, можно переходить к проектированию технологии. И на основе данных, полученных о пользователях создается настолько удобная и интуитивно понятная система, на сколько это возможно. После запуска уже готовой технологии наступает этап оценки. Оценивается не безопасность либо работа самой системы, а то, насколько людям удобно пользоваться ей на практике. Если процесс использования системы оказывается слишком сложным, то люди найдут способы обхода защиты, либо будут искать ее альтернативы. Для того, чтобы убедится в грамотности работы, необходимо проводить оценку системы регулярно [1].

Одним из популярных способов оценки работы системы на примере многопользовательского веб сайта является A/B тест. Суть данного эксперимента состоит в том, что в первоначальный дизайн сайта, который предложен пользователям, вносятся некие изменения. К примеру, меняется положение определенных блоков на странице, меняются некоторые цветовые элементы, меняется форма кнопок, или вносятся изменения в саму структуру страницы. Таким образом пользователям предлагается два вида веб страницы, которые выпадают в случайном порядке, т.е. половине от пользователей сайта

выпадает дизайн A, а другой половине - дизайн B.

Рис. 1. *Иллюстрация A/B тестирования*

Данные эксперимента для обоих вариантов сайта обрабатывается по нескольким параметрам, таким, как конверсия, время, проведенное пользователями на странице, для обоих вариантов дизайна по отдельности, количество переходов используя элементы сайта (кнопки, рекламные блоки), и просмотры отдельных страниц. A/B тест не может проводится на основе наших предположений и догадок. Результат теста должен быть основан исключительно на данных полученных в ходе исследования, после чего уже можно установить какой из предложенных вариантов дизайна оказался более удобным и интуитивно понятным для пользователей [2].

Одним из основных аспектов для пользователя является удобство использования – способ проверки и понимания насколько удобно людям пользоваться системой на практике (*англ.* usability). Компоненты проверки юзабилити можно разделить на несколько ключевых аспектов:

Скорость – время, которое необходимо пользователю для выполнения определенных действий. Чем меньше времени затрачивает пользователь, тем система считается более удобной для эксплуатации.

Рассмотрим данный механизм на примере снятия блока экрана на мобильном телефоне. Сравнивая время для снятия блока с помощью отпечатка пальца и ввода пароля, достигается следующий результат: в случае отпечатка пальца, время снятия блока составляет примерно 1 секунду, а в случае ввода пароля вручную примерное время

разблокировки составит 4 секунды.

Рис. 2. *Сравнение типов блокировки экрана*

Измеряется эффективность данного метода с помощью времени, затраченного на выполнение задачи. Следовательно, по параметру скорости, использование отпечатка

пальца, по сравнению с вводом пароля вручную, для пользователей является более удобным вариантом.

Продуктивность – количество ошибок, допущенных до выполнения задачи. В процессе даже самой банальной операции, такой как снятие блока с экрана мобильного телефона, можно столкнутся с рядом неверных вводов отпечатка пальца / пароля. В результате пользователь видит сообщение об ошибке, и в некоторых случаях подсказку для дальнейших действий, что само по себе является отдельных элементом в спектре удобства использования (usability). Данном элементе описан более подробно ниже.

***Рис. 3**. Иллюстрация вывода ошибок системой*

Продуктивность системы определяется подсчетом ошибок до выполнения задачи. Чем меньше ошибок делается в ходе выполнения того или иного действия, тем система в аспекте продуктивности является более удобной для пользователя. Сравнивать методы ввода пароля и отпечатка пальца по данному параметру нельзя, так, как количество ошибок при выполнении задачи может меняться в зависимости от целого ряда факторов, таких как среда, в которой происходит действие, навыки пользователя и другие.

Усваиваемость – как быстро пользователь учится работать с системой. В данном аспекте для упрощения изучения системы пользователем задействованы такие методы как

подсказки, либо указатели для дальнейших действий. Использование данного метода на практике можно наблюдать к примеру, в мобильных операционных системах [3].

Рис. 4. *Иллюстрация подсказок системы*

Измеряется данный аспект сравнением времени, которое тратит пользователь на выполнение поставленной задачи в первый раз, и времени, которое ему понадобится на выполнение той же самой задачи во второй, третий, четвертый раз. Каждое повторное выполнение задачи должно занимать меньше времени, чем предыдущее (первый вход - 10сек, второй вход - 8 сек, третий - 7сек.)

Запоминаемость – после того, как пользователь освоил систему, сколько ему потребуется времени для повторной работы с той же системой, но после длительного промежутка времени. И насколько удобно ему будет пользоваться этой системой в будущем. На практике данный метод используется во многих программах, таких как MS Word, либо других редакторах контента. Этот метод основан на зрительной памяти человека, в программе в графе выбора шрифта, с которым пользователь хочет работать, помимо названия самого шрифта, приведен так же пример это отображения.

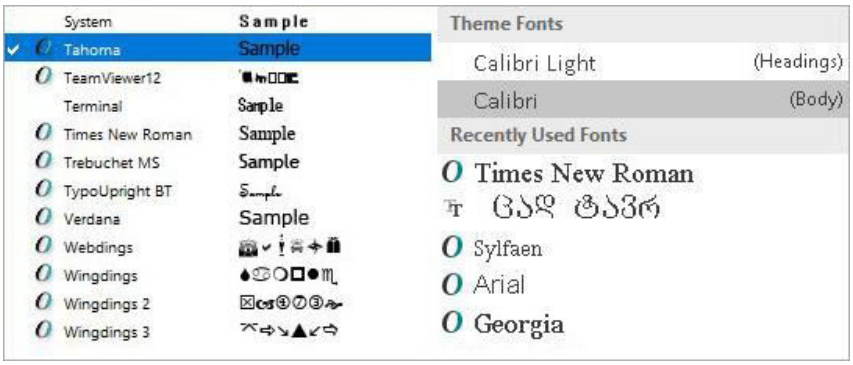


Рис. 5. *Демонстрация визуальной части для каждого шрифта*

В некоторых программах данный вариант представлен в виде названия шрифта, написанного этим самым шрифтом. Измерить работу данного метода на практике можно с помощью замера времени, которое в будущем потребуется пользователю на то, чтобы вспомнить, как работать с данной системой.

Предпочтения пользователей – что пользователям понравилось больше. Данный метод основан на предпочтениях пользователей, которые и диктуют направления в развитии удобства использования системы. Измерить данный компонент можно с помощью групповых дискуссий и опросов пользователей.

Суммируя все приведенные выше примеры, мы делаем вывод, что для разработки грамотной и удобной многопользовательской системы необходимо учитывать целый ряд факторов и методов взаимодействия человека с компьютером. Одним из ключевых моментов является безопасность самой системы, и в ее рамках надо находить оптимально удобные способы работы для пользователей [4]. Только с учетом практических аспектов и примеров использования данных методов на практике, можно создавать действительно удобный инструмент для большого количества пользователей. Еще одним важный факт - описанное нами выше удобство использования, которое является базой для разработки системы. А разрабатывая систему можно только с учетом реального опыта и предпочтений пользователей. Так же надо отметить важность регулярной оценки уже готового продукта, и проведений постоянной работы над улучшением и оптимизацией системы максимально учитывая потребности и возможности пользователей.

Библиографический список:

1. Hau San Wong, Horace H. S. Ip. - Human Computer Interaction // Encyclopedia of Multimedia pp 289-293. 2008
2. R. Kohavi, S.Thomke - A/B Testing: How to get in right // Harvard Business Review. 2017
3. H. Almut Usable Security Policies for Runtime Environments // Institutionen för datavetenskap , 2007.
4. M.Iavich, G.Iashvili CAPTCHA analysis and its problems // Scientific & practical cyber security journal (SPCSJ) № 1 .[Electronic journal]. URL: http://journal.scsa.ge/issues/2017/09/415

Scientific and Practical Cyber Security Journal (SPCSJ) 1(2):54-56
Scientific Cyber Security Association (SCSA), 2017 ISSN: 2587-4667

DANGERS OF USING BANK CARDS WITH CONTACTLESS PAYMENT TECHNOLOGY (PAYPASS, PAYWAVE AND OTHERS)

V.Ivanov, B,Horbatenko, V.Stopin

Kharkiv National University of Radio Electronics (NURE), «Computer science» faculty, department of «System engineering»

ABSTRACT:

Not so far ago, the world saw such a phenomenon as «contactless payments». Bank card, which has PayPass chip inside, proved to be comfortable in everyday using. There is no need to put the card into ATMs or terminals as well as to remember and enter PIN-code, to fill receipts and many other routine staff, which people do not like doing. Developers of this technology guarantee that cards with chip are totally secure and safe payment method, however, is it really so? In this article let's answer on this and some other questions about bank cards with a chip.

KEYWORDS: PayPass, payWave, cyber, security, theft, payment, visa, mastercard

Не так давно свет увидел такое явление, как «бесконтактные платежи». Банковская карта, в которую внедрен чип PayPass, показала себя удобной в использовании. Нет необходимости вставлять карту в банкомат или терминал, помнить и вводить PIN-код, заполнять чеки, и многие другие рутинные занятия, которые не нравятся людям в повседневной жизни. Разработчики данной технологии уверяют, что карта с чипом – абсолютно безопасное средство оплаты, однако, так ли это на самом деле? В этой статье ответим на этот и ряд других вопросах о банковских картах с чипами.

Данная статья несет исключительно ознакомительный характер, а также мероприятия, позволяющие противодействовать злоумышленникам. Никогда не пытайтесь повторить то, что описано в данной статье.

Бесконтактные банковские карты с технологией PayPass (или ее аналогами) используют технологию NFC для передачи данных, а NFC – это разновидность RFID. На карте с данной технологией находится чип и антенна, которые «отвечают» на запросы платежного терминала на радиочастоте 13,56 МГц. Существует множество стандартов бесконтактных

платежей, среди них: Visa payWave, MasterCard PayPass, American Express ExpressPay и так далее, но все они устроены похожим образом и не имеют кардинальных различий в структуре своей работы.

Расстояние, на котором данные карты функционируют и отправляют запросы, составляет около 3-5 см от считывающего устройства. То есть, приложив к терминалу карту на данном расстоянии, вы автоматически оплачиваете покупку без ввода пин-кода и каких-либо дополнительных проверок. Но стоит обратить внимание, что дополнительных проверок не последует, при условии, если не превышается лимит, установленный на сумму оплат посредством бесконтактного платежа. Для разных стран имеются собственные лимиты. Таким образом, в России, по умолчанию доступно ~85$, в Украине – ~38$, США – ~50$. Если указывать сумму для оплаты ниже обозначенного лимита, то никаких подтверждений не требуется.

Злоумышленник, желая заполучить чужие средства, может поступить следующим образом: происходит покупка фиктивной компании или фирмы на фиктивное лицо, обязательным условием является наличие расчетного счета в банке у этой компании. Каким образом это производиться в наше время, и какими средствами, в данной статье описано не будет, но знайте, что выполнить данный пункт очень легко [1]. Далее, злоумышленник приобретает беспроводное средство для проведения платежей (платежный терминал, эквайринг), имеющее доступ к сети, для проведения платежей. Это может быть, как мобильный интернет, так и любые другие средства, которые позволяют получить бесперебойный доступ к сети Интернет. Существует множество модификаций платежных терминалов, у которых усилена антенна передачи данных, более мощный сигнал связи и так далее.

Затем, злоумышленник отправляется в массовое скопление людей (например, торговый центр), где находятся его потенциальные жертвы. Сумма для оплаты в терминале программируется заранее. Злоумышленнику лишь остается положить терминал в тонкую сумку и приблизиться к месторасположению кредитной карты жертвы на расстояние до 5 см. Если карта находится достаточно близко к терминалу, то происходит списывание средств, о чем свидетельствует вибрация или специальный сигнал, что дает понять злоумышленнику, что операция прошла успешно.

Многие банковские системы не предусматривают защиты от повторных транзакций. Таким образом, недобросовестный человек может произвести несколько транзакций с небольшим промежутком времени. При этом, никаких дополнительных проверок или запросов PIN-кода не будет.

Как итог, необходимо обозначить, что использование карт с технологией бесконтактных платежей не является безопасным. Злоумышленник, в примечании к платежу, может написать что-либо непримечательное, обыденное для пользователя. Например, это может быть «обслуживание карты», при этом указывается небольшая сумма, чтобы тем самым не вызвать каких-либо недоразумений при проверке платежей. К сожалению, банковские системы не спешат исключать данные карты из общего пользования. Поэтому необходимо максимально обезопасить себя от возможных попыток списания средств [2]. Следующие советы помогут снизить вероятность списания личных средств с ваших карт без вашего ведома:

1) Храните бумажник в труднодоступных местах.
2) Создайте экранирование вокруг карты, например, обмотайте её фольгой. Она не пропустит сигнал терминала. Таким образом, злоумышленник не сможет выполнить запрос о снятии средств.
3) Приобретите специальный экранированный кошелек.
4) Будьте бдительны и обращайтесь в ваш банк при малейшем подозрении о несанкционированном списании средств.
5) Активируйте моментальное оповещение о транзакциях с вашей картой (смс оповещения, звонки и другое).
6) Измените максимальный лимит транзакции на минимально возможный и приемлемый для вас.
7) Избегайте близкого контакта с людьми имеющие тонкие сумки.
8) Поставьте обязательный ввод пин-кода, если это позволяет ваш банк.
9) Держите вашу кредитку в поле вашего зрения при оплате на кассе или где-то еще.

Данные советы не гарантируют полной безопасности, однако при их соблюдении вы значительно снижаете вероятность мошенничества и проведения несанкционированных платежей. По возможности исключите из использования подобные банковские карты. Не экономьте время, потраченное на подтверждение оплаты услуг, берегите личные средства.

Библиографический список:

1. Cybersecurity and Cyberwar: What Everyone Needs to Know® 1st Edition by P.W. Singer, Allan Friedman
2. Social Engineering: The Art of Human Hacking 1st Edition by Christopher Hadnagy, Paul Wilson

Scientific and Practical Cyber Security Journal (SPCSJ) 1(2):56-60
Scientific Cyber Security Association (SCSA), 2017 ISSN: 2587-4667

ON DIGITAL SIGNATURE SCHEMES

[1]N. Inassaridze, [2]M. Joglidze

[1] A.Razmadze Mathematical Institute of Tbilisi State University, Tamarashvili Str. 6, Tbilisi 0177, Georgia & Georgian Technical University & Tbilisi Centre for Mathematical Sciences
[2]University of Georgia, Kostava Str.77a, Tbilisi 0171, Georgia

ABSTRACT.

Digital signature schemes are fundamental cryptographic primitives, useful as a stand-alone application, and as a building block in the design of secure protocols and other cryptographic objects. In this article, we give general overview of basic notions of digital signature schemes and discuss the multiple-time digital signature scheme given in [8].

1. INTRODUCTION. Nowadays digital signatures have become a key technology for making the Internet and other IT-infrastructures secure. Instead of outdated traditional physical signatures, digital signatures are turning more important tools to implement secure and correct signs. Providing authenticity, integrity, and non-repudiation of data, digital signatures are widely used in identification and authentication protocols. Hence, the existence of secure signature algorithms is crucial for maintaining IT-security. The digital signature algorithms that are used in practice today are RSA [11], DSA [2], and ECDSA [5]. They are not quantum immune since their security relies on the difficulty of factoring large composite integers and computing discrete logarithms.

Hash-based digital signature schemes offer a very promising alternative to RSA and elliptic curve signature schemes, which were invented by Ralph Merkle [7]. Merkle started from fundamental, one-time signature schemes from [6]. One-time signature schemes proposed by Lamport-Diffie [6] and Rabin [9] were among the earliest signatures based on the idea of committing to private keys by one-way functions (see Rompel [12]). A severe disadvantage of these schemes is that one key-pair can only be used to sign and verify a single document, hence they are inadequate for most applications. By this reason multiple-time signature schemes are invented, that can be used to sign a predetermined number of messages [7, 8].

Despite the limit imposed on the number of messages signed, multiple-time signatures are very interesting cryptographic primitives as they typically offer more efficient generation and verification of signatures than the schemes based on public-key cryptography, and typically are constructed based on an arbitrary one-way function without requiring a trapdoor function.

This article is an expository article about digital signature schemes, while particular attention is paid to multiple-time signature schemes. In Section 2 we discuss the basic notions of digital signature schemes. In Section 3 we draw attention to the multiple-time digital signature scheme HORS++ from [7], which will be the subject of our further investigation and an important component in the construction of a new hybrid multi-time post-quantum signature scheme.

2. FUNDAMENTAL NOTIONS. In this section we discuss digital signature schemes in general manner and recollect their basic fundamental notions (see e.g. [4]). Dealing with digital signatures one should know the following three basic aspects:

(1) The essence of a digital signature scheme;
(2) The types of attacks the adversary is able to mount against a digital signature scheme;
(3) The meaning of "breaking" a digital signature scheme.

2.1. Digital signature schemes. A digital signature schemes in its standard form consists of the following parts:

- A security parameter k, which is chosen by the user when he creates his public and secret keys. This parameter determines a number of quantities (length of signatures, running time of the signing algorithm, length of signable messages, etc).
- A message space M, which is the set of messages to which the signature algorithm may be applied, is assumed to consist of binary strings, i.e. $M \subseteq \{0,1\}^*$. To ensure that the entire signing process is polynomial in the security parameter, the length of the messages is supposed to be bounded by k^c, for some constant $c > 0$.
- A signature bound B, which is an integer bounding the total number of signatures that can be produced with an instance of the signature scheme. This value may be infinite, though it is typically bounded above by a low-degree polynomial in k.
- A key generation algorithm G, which any user can use on input 1^k to generate in polynomial time a pair (K_{priv}, K_{pub}) of matching private, sometimes called the trap-door information, and public keys.
- A signature algorithm σ, which produces a signature $\sigma(m, K_{priv})$ for a message $m \in M$ using the private key K_{priv}. σ may receive other inputs too.
- A verification algorithm V, which tests whether S is a valid signature for a message m using the public key K_{pub}. It means, that $V(S, m, K_{pub})$ is true iff it is valid.

Any of the above algorithms may be *randomized* algorithms that make use of auxiliary random bit stream inputs. We should note that G must be a randomized algorithm, since part of its output is the secret key, which must be unpredictable to an adversary.

2.2. Types of attacks. It is distinguished two basic types of attacks to a digital signature scheme. There is an attack in which the adversary knows only the real signer's public key. This type of attack is called *key-only attack*. In another type of attack the adversary can examine some signatures corresponding to known or chosen messages before his attempt to break the scheme. This type of attack is called *message attack.*

Four kinds of message attacks are identified, divided according to how the messages whose signatures the adversary sees are chosen. Denote by A a user whose signature method is being attacked. Namely,

Known-message attack. The adversary has access to signatures for a set of t (known to him) messages $m_1,\ldots, m_t$, which are not chosen by him.

Generic chosen-message attack. The adversary can obtain valid signatures from A for a list of messages $m_1,\ldots, m_t$, chosen before he attempts to break A's signature scheme (nonadaptive attack). These messages are chosen by the enemy, but they are fixed and independent of A's public key (generic attack).

Directed chosen-message attack. This is similar to the generic chosen-message attack, except that the list of messages to be signed may be created after seeing A's public key but before any signatures are seen. This attack is directed against a particular user A but is still nonadaptive.

Adaptive chosen-message attack. The adversary can request from *A* signatures of messages that depend not only on public key but that depend additionally on previously obtained signatures.

2.3. "Breaking" of a digital signature scheme. It is said that the adversary has "broken" user *A*'s signature scheme if his attack allows him to do any of the following with a non-negligible probability:

Total break. The adversary computes *A*'s secret trap-door information.

Universal forgery. The adversary finds an efficient signing algorithm functionally equivalent to *A*'s signing algorithm, based on possibly different but equivalent trap-door information.

Selective forgery. The adversary forges a signature for a particular message chosen a priori by him.

Existential forgery. The adversary forges a signature for at least one message, which is not controlled by him, so it can be random or nonsensical.

A scheme is respectively totally breakable, universally forgeable, selectively forgeable or existentially forgeable if it is breakable in one of the above senses. Note that it is more desirable to prove that a scheme is not even existentially forgeable than to prove that it is not totally breakable.

3. MULTIPLE-TIME DIGITAL SIGNATURE SCHEME. In this section we discuss the digital signature scheme of Pieprzyk, Wang and Xing [8], called HORS++, and generalizing the one-time signature scheme previously proposed by Reyzin and Reyzin [10]. The HORS++ scheme can be used to sign predetermined number of messages. To construct the scheme, a well-known combinatorial object, called the cover-free family is used, which is introduced by Erdös et al [3]. Let *[t]* denote the set of first *t* natural numbers, *{1,2,..., t}*.

Definition. A pair of sets *([t], B)* with $B = \{B_i \subseteq [t] \mid i = 1, \dots, n\}$ is called an (n, t, r)-cover-free family if for any subset $\Delta \subseteq \{1,\dots n\}$ with $|\Delta| = r$ and any $i \notin \Delta$,

$$\left|\boldsymbol{B_i} \setminus \cup_{\boldsymbol{j}\in\boldsymbol{\Delta}} \boldsymbol{B_j}\right| \geq \mathbf{1}.$$

Now suppose that *([t], B)* is an *(n, t, r)*-cover-free family and $g:\{0,1\}^* \longrightarrow \{0,1\}^b$ a cryptographic hash function with $2^b \leq n$. Suppose also that $S: \{0, 1\}^b \longrightarrow B$ is an injective mapping and $f:\{0,1\}^l \longrightarrow \{0,1\}^l$ a one-way function operating on *l*-bit strings, for a security parameter. The HORS++ scheme works as follows.

Key generation. For the given security parameter 1^l, the private key of the scheme k_{priv} consists of random *t* bit strings of length *l*

$k_{priv} = (s_1, \dots, s_t)$.

Then the public key of the scheme is

$k_{pub} = (v_1, \dots, v_t)$,

where any $v_i = f(s_i)$, $i \in [t]$.

Signature generation. A message $m \in \{0,1\}^*$ is signed using the pivate key k_{priv}. At first the message digest $g(m) \in \{0,1\}^b$ of *m* is computed. Next the mapping S is used to compute $S(m) = \{i_1, \dots, i_k\} \in B$. Then the signature of the scheme is $(s_{i_1}, \dots, s_{i_k})$.

Signature verification. To verify a signature $(s'_1, \ldots, s'_k)$ on a message m, again it is calculated $S(m) = \{i_1, \ldots, i_k\} \in B$. Finally, it is checked whether
$(f(s'_1), \ldots, f(s'_k)) = (v_{i_1}, \ldots, v_{i_k})$.

As we already mentioned, the HORS++ scheme generalizes the one-time signature scheme proposed by Reyzin and Reyzin in [10], if $n = \binom{t}{k} \geq 2^b$ for some integer $k > 0$, $r = 1$ and (n, t, r)-cover-free family B is the set of all k-subsets of [t]. Note also that the scheme of Reyzin and Reyzin is a simple generalization of that given by Bos and Chaum [1], if $t=2k$. The scheme of Bos and Chaum is in turn a generalization of the scheme of Lamport and Diffie [18], if $b = k$ and the mapping S is given by the algorithm: for any bit string $m = (m_1, \ldots, m_k)$ of length k, compute $S(m)$ as $\{1+m_1, \ldots, 2k-1+m_k\}$.

Security. Suppose that the adversary has seen r valid signatures for the messages $m_1, \ldots, m_r$ chosen adaptively. In order to forge a signature on a new message, the adversary would have to invert the one-way function f on the value associated to the points of $S(m_{r+1}) \setminus \bigcup_{i=1}^{r} S(m_i)$ in the public key. Since $([t], B)$ is an (n, t, r)- cover-free family, it yields that $|S(m_{r+1}) \setminus \bigcup_{i=1}^{r} S(m_i)| \geq 1$. That means that the adversary has to invert the one-way function on at least one value, and so the security of the signature is reduced to the one-wayness of f.

Efficiency. To measure the efficiency of the scheme, it is considered two aspects of performance:

(i) the time needed for key generation, signing, and verifying;

(ii) the length of secret key, public key, and signature.

The key generation requires t evaluations of one-way function, the signing takes as long as the running time of the algorithm for S and the verifying algorithm takes the same time as signing, plus at most t evaluations of the one-way function. The size of public and secret key is determined by t and the size of signature is determined by the size of blocks $|B_i|$ in B. Thus, the performance of the HORS++ scheme is determined by the parameters of the underlying cover-free family. One has the following

Theorem [8]. Given a one-way function f with the l-bit input and f_l -bit output. There exists a r-time signature scheme secure against the adaptive chosen-message attack with the secret key size $O(r^2 f_l l)$-bits, public key size $O(r^2 f_l^2)$-bits, and with the size of signature $O(r f_l l)$.

However, without taking into account the complexity of the mapping S, this theorem has only theoretical interest of its existence. Implementation of the mapping S is the most time consuming part of the system. To make the given HORS++ scheme practical, in [8] some algorithms of the mapping S are proposed based on polynomials, error correcting codes and algebraic curves.

ACKNOWLEDGMENTS

The first author was supported by STCU-2016-08/MTCU 6321 and by the Agencia Estatal de Investigacííon, Spain (European ERDF support included, UE) grant number MTM2016-79661-P.

REFERENCES

[1] J.N.E.Bos and D.Chaum, Provably unforgeable signature, Advances in Cryptology – Crypto'92, LNCS, 740 (1993), 1-14.

[2] T.ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithms, Advances in Cryptology – CRYPTO '84, LNCS 196, Springer (1985), 10-18.

[3] P.Erdös, P.Frankl, and Z.Furedi, Families of finite sets in which no set is covered by the union of r others, Israel Journal of Mathematics, 51 (1985), 79-89.

[4] S.Goldwasser, S.Micali and R.Rivest, A digital signature scheme secure against adaptive chosen-message attacks, SIAM J. Comput. 17(2) (1988), 281-308.

[5] D.Johnson and A.Menezes, The elliptic curve digital signature algorithm (ECDSA), Technical Report CORR 99-34, University of Waterloo, 1999. Available at http://www.cacr.math.uwaterloo.ca.

[6] L.Lamport, Constructing digital signatures from a one way function, Technical Report SRI-CSL-98, SRI International Computer Science Laboratory, 1979.

[7] R.C.Merkle, A certified digital signature. Advances in Cryptology – CRYPTO '89 Proceedings, LNCS 435, Springer (1989), 218-238.

[8] J.Pieprzyk, H.Wang and C.Xing, Multiple-time signature schemes against adaptive Chosen Message Attacks, In: Matsui, M., Zuccherato, R. (eds.) SAC 2003. LNCS 3006 (2004), 88-100.

[9] M.O.Rabin. Digitalized signatures, Foundations of Secure Communication, Academic Press (1978), 155-168.

[10] L.Reyzin and N.Reyzin, Better than BiBa: Short one-time signatures with fast signing and verifying, Information Security and Privacy (ACISP02), LNCS 2384, 144-153.

[11] R.L.Rivest, A.Shamir and L.Adleman, A method for obtaining digital signatures and public-key cryptosystems, Communications of the ACM, 21(2) (1978), 20-126.

[12] J.Rompel, One-way functions are necessary and sufficient for secure signatures, Proceedings of ACM STOC'90 (1990), 387-394.

Scientific and Practical Cyber Security Journal (SPCSJ) 1(2):60-68
Scientific Cyber Security Association (SCSA), 2017 ISSN: 2587-4667

CRITICAL ANALYSIS OF SOME CRYPTOGRAPHY ALGORITHMS

E. Jincharadze

Georgian Technical University

ABSTRACT

Nowadays large amounts of data are being transferred over different network channels that could be both public and private. Protecting information is more vital than ever. One of the main problems during transferring data is to ensure security. Nowadays the exchange of valuable information over Internet, such as bank transactions, credit card numbers and telecommunication services are already common practices. Because the world becomes more connected and communication methods are developed, the security on electronic services has become more important. In order to protect valuable data in computer and communication systems from unauthorized attack and modification different security methods must be involved. Cryptography is one such method to make sure that confidentiality, authentication, integrity, availability and identification of user data can be secured. Cryptography provides security and privacy of used data. Encryption is the process of converting normal data or plaintext to something incomprehensible or cipher-text by applying mathematical transformations or formulae. These mathematical transformations or formulae used for encryption processes are called algorithms.

At the present time, cryptography plays important role to provide security communication between multiple objects. In many modern studies, researchers are trying to identify best cryptography mechanisms with their strong and weak points in terms of their performance results. To select cryptographic technique according to a particular situation is not so easy task. To solve this issue we have to understand that technique selection is totally dependent on desired quality attributes such as efficiency and security.

In this paper is presented critical analyze of some cryptography algorithms DES, 3DES, AES, Blowfish and RSA is presented. According to the literature review we have analyzed the performance and efficiency of those algorithms. To make an evaluation of those systems was explained imitations of sample context. We have analyzed various encryption algorithms on the basis of different parameters and compared them to choose the best data encryption algorithm so that we can use it in our future work.

Introduction

Cryptography is the discipline that studies the mathematical techniques which are related to information security such as providing the security services of confidentiality, data integrity,

authentication and nonrepudiation. The art and science of keeping messages secure is cryptography, and it is practiced by cryptographers [1]. More generally, cryptography algorithm is the technique or some formula that makes data or network secure by providing security. Cryptosystems are complex combinations of hardware and software used to transform plaintext messages into a series of unintelligible characters known as cipher text, then back to their original plaintext known as cipher text, then back to their original plaintext form. "An encryption algorithm scrambles data by combining the bits in the key with the data bits; in decryption, the algorithm unscrambles data by separating the data bits from the key bits." [3]. Cryptography is the science which ensures that information is sent in such secure way that the only person able to retrieve this information is the intended recipient. Contemporary cryptography is unable to meet the requirements, in that its use would impose such severe inconveniences on the system users, as to eliminate many of the benefits of teleprocessing [6].

Cryptography falls into two important categories: secret and public key cryptography. Both categories play their vital role in modern cryptographic applications. For several crucial applications, a combination of both secret and public key methods is indispensable [7].

There is some basic terminology used in cryptography, which we should know for better understanding of encryption algorithms. This terminology is very important to understand because in every algorithm description. Those common terms are**: Plain Text** - The original text or message used in communication in called as Plain text. **Cipher Text -** The plain text is encrypted in un-readable message. This meaningless message is called Cipher Text. **Encryption -** is a process to convert Plain text into Cipher text. This converted text can securely be transferred over the unsecure network. Encryption process is done using encryption algorithm. **Decryption -** Decryption process is the reverse of Encryption process. So we have simple function E(M)=C, D(C)=M, D(E(M))=M. **Key** is a numeric or Alpha-numeric text (mathematical formula). In encryption process it takes place on Plain text and in decryption process it takes place on cipher text. **Key size** is the measure of length of key in bits, used in any algorithm. **Block Size** - Key cipher works on fixed length string of bits. This fix length of string in bits is called Block size. This block size depends upon algorithm. **Round** of encryption means that how much time encryption function is executed in complete encryption process till it gives cipher text as output.

Cryptography has some goals that need to be ensured for user's information security. Modern cryptography concerns itself with the following four services. Let us now see the possible goals intended to be fulfilled by cryptography. **Confidentiality** is the fundamental security service provided by cryptography. It is a security service that keeps the information from an unauthorized person. It is sometimes referred to as privacy or secrecy. **Data Integrity** It is security service that deals with identifying any alteration to the data. The data may get modified by an unauthorized entity intentionally or accidently. Integrity service confirms that whether data is intact or not since it was last created, transmitted, or stored by an authorized user. **Authentication** provides the identification of the originator. It confirms to the receiver that the data received has been sent only by an identified and verified sender. **Non-repudiation -** It is a security service that ensures that an entity cannot refuse the ownership of a previous commitment or an action. It is an assurance that the original creator of the data cannot deny the creation or transmission of the said data to a recipient or third party.

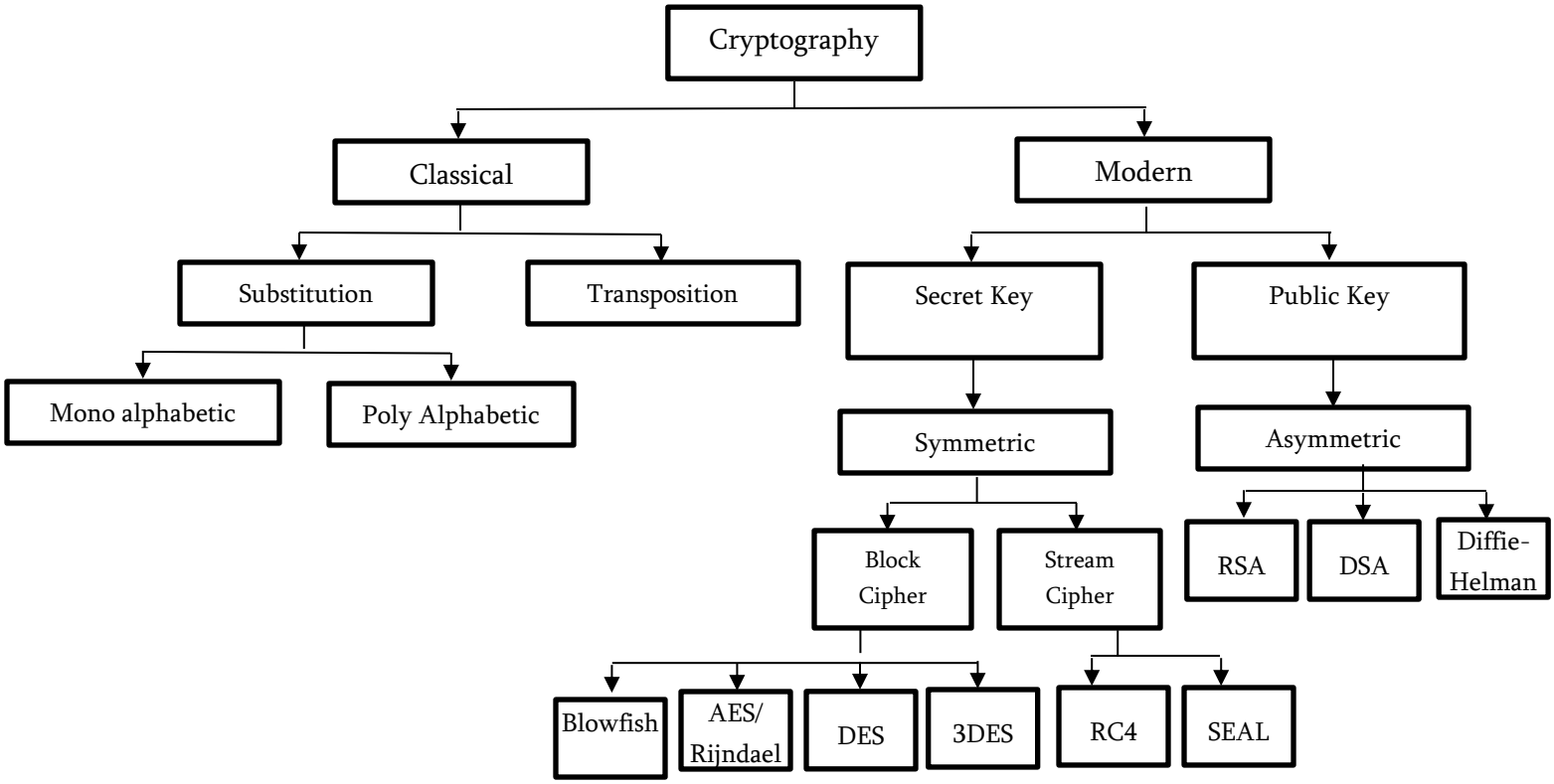


Figure 1. General overview of cryptographic techniques.

There are number of different encryption techniques which can be broadly divided into two categories: Symmetric Key Encryption and Asymmetric Key Encryption.

Symmetric algorithms, sometimes called conventional algorithms, are algorithms where the encryption key can be calculated from the decryption key and vice versa. In most symmetric algorithms, the encryption key and the decryption key are the same. These algorithms, also called secret-key algorithms, single-key algorithms, or one-key algorithms, require that the sender and receiver agree on a key before they can communicate securely [1]. Encryption and decryption with a symmetric algorithm are denoted by: $E_K(M) = C$, $D_K(C) = M$. In Symmetric key Encryption same key is used to encrypt and decrypt data. Receiver uses the same key and the corresponding decryption algorithm to decrypt the data. At the same time symmetric key cryptography is classified into two categories –Stream Ciphers and Block Ciphers. A stream cipher breaks the plaintext M into consecutive characters or bits, so we have following consequence m_1 , m_2 ,.. m_i. And each m_i is encrypted with k_i key, where $K= k_1 , ... , k_i$. So we have following equation $E_K(M)=E_{k1}(m_1)\ E_{k2}(m_2) \ldots E_{ki}(m_i)$.

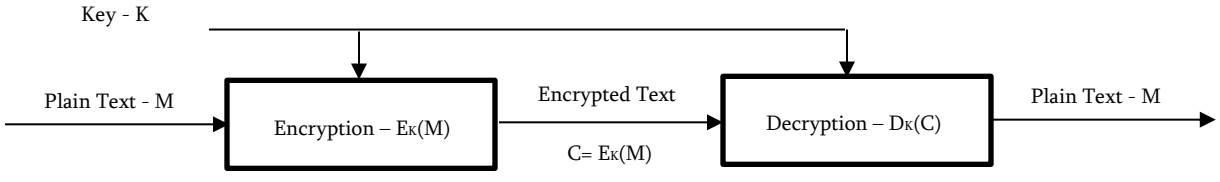


Figure 2. Symmetric key Cryptography

Asymmetric Key Encryption uses two different keys: public and private keys. Public key for encryption purpose and private key for decryption. Asymmetric Cryptography is cryptographic system which requires two separate keys K_E and K_D, where $K_E \neq K_D$. K_E is user to encrypt the plaintext - M, and K_D is used to decrypt the cipher text - C. Encryption key is published or public and the decryption key is kept private. Public key algorithms, unlike symmetric key algorithms, do not require a secure initial exchange of secret keys between the parties [2]. Public-key cryptography is a method which assures the confidentiality, authenticity of digital communications and data storage.

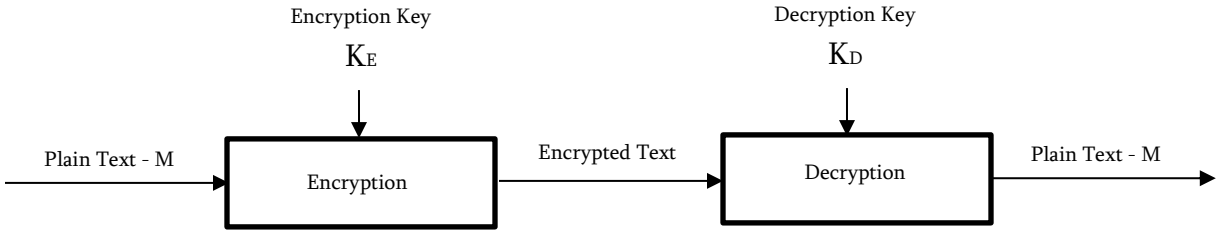


Figure 3. Asymmetric key Cryptography

OVERVIEW OF SOME CRYPTOGRAPHIC ALGORITHMS

There are many different types of cryptography algorithms. In this topic is described some of them and is analyzed each of them by terms of their usage and vulnerability. Is analyzed those algorithms performance for their evaluation. All of these algorithms are unique on it's way. However, the problem is that how to find the best security algorithm which provides the high security and also take less time for a key generation, encryption, and decryption of information. Security algorithms will depend on pros and cons of each algorithm, requirement and suitable for different application [13].

Data Encryption Standard (DES) **–** is a symmetric (secret key) block cipher algorithm, which was published as FIPS-46 in the Federal Register in January 1977 by the National Institute of Standards and Technology (NIST). DES is a block cipher that enciphers 64-bit blocks of data with a 56 bit key. The other 8 bits are used for checking correspondence. DES algorithm for decryption uses the same structure as encryption but with the keys used in reverse order. This is the advantage of DES algorithm because the same hardware or software can be used in both directions. But because of short key length DES is weaker about attacks. DES was approved in 1974 since that time many attacks were reported, which has made DES as insecure algorithm, so that DES could be broken under a known-plaintext attack by exhaustive search. It was also experienced that a special purpose machine consisting of a million LSI chips could try all $256 \approx 7$ X 10 16 keys in 1 day [1][3][7]. DES is not an ideal encryption technique in modern cryptography, instead it is used in mode of

operation. The key length of DES system is 56 bits, that means that only 56 bits are actually used in the algorithm. So it means that we need maximum of 2^{56} attempts to find the correct key [14].

The weak point on DES attack is Brute force attack. DES' another weak point is it's slow encryption speed.

Triple-DES (3DES) **–** is one of the block cipher methods of symmetric key cryptography. 3DES was designed to improve weak points of DES but it is not totally different form DES. Triple DES is considered to be DES- three times. 3DES increases the key size from 56 bits to 168 bits to make it resistant from brute-force attack. It has two variants: one with two keys and other with three keys.

The negative side of 3DES is that it is slower than other block encryption methods, but it is more secured because of longer key length as it reduces many attacks. The strong side of 3DES algorithm is that it is three times secured than DES that's why it is better than DES encryption algorithm. 3DES provides security to the data but it is not the best because it consumes lot of time.

Advanced Encryption Standard (AES) **-** is a symmetric block cipher, is also known as Rijndael algorithm and it was developed by NIST in late 90's. Was developed to protect classified information and is implemented in software and hardware throughout the world for sensitive data encryption. AES is actually, three block ciphers, AES-128, AES-192 and AES-256. 56-bit key of DES was not safe against the brute force attack and 64-bit blocks and was weak. AES encrypts data blocks of 128 bits using variable key length of 128,192 or 256 bits in 10, 12 or 14 rounds depending upon the key size [1] [3] [4]. In Advanced encryption standard there are 10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys[21]. The AES algorithm holds a 4 by 4 array of bytes called the state, that is initialized to the input of 128 bits (i.e., 16 bytes) to the cipher. The substitution and permutation operations are all applied to the state array. Each round of AES consists of four operations. AES can be implemented in small devices for encrypting a message to send over a network. Some other applications are monetary transaction and security applications [13] [16].

Blowfish is a symmetric block cipher algorithm. Blowfish uses the same secret key to both encryption and decryption of messages. Blowfish has block with 64 bits size. It uses a variable – length key, from 32 bits to 448 bits. Blowfish was developed by Bruce–Schneider in 1993 as an alternative to the existing encryption algorithms. It is appropriate for applications where the key is not changed frequently. It has 16 or less rounds. It is considerably faster than most encryption algorithms when executed in 32-bit microprocessors with huge data caches.

Blowfish provides good encryption and for this moment there is not known any attack to be successful against Blowfish. It is much faster than DES. But weak point of Blowfish algorithm is the weak key.

Rivest-Shamir-Adleman Algorithm – RSA is an asymmetric cryptographic algorithm which is used for encryption and decryption of the plain text. RSA is usually used in transferring of keys over

an insecure channel. Because of that RSA is asymmetric algorithm, there are two keys used in the algorithm. In RSA, the encryption key is public and differs from the decryption key which is kept secret [1]. Messages encrypted with the public key can only be decrypted in a reasonable amount of time using the private key. RSA provides confidentiality, integrity, authenticity, and nonrepudiation of data [1] [2]. RSA is more commonly used in electronic industry for online money transfer [19].

In RSA cryptographic algorithm the main disadvantage is its encryption speed, because for encryption it consumes lot of time. Generally this is disadvantage of asymmetric key algorithms because the use of two asymmetric keys. RSA provides good level of security but the same time it is slow for encrypting files. Another weak point of this algorithm is usage of fake key at decryption level. For successful encryption the secret key should be private and correct.

LITERATURE SURVEY

According to the different literature survey we could say that encryption algorithms AES showed poor performance results compared to other algorithms as it requires more processing power. The comparison also reveals that 3DES requires always more time than DES because of its three phase encryption technique.

Hamdan O. Alanazi made a comparison of three symmetric block ciphers DES, 3DES and AES on nine parameters. They got following result DES is vulnerable to brute force attack and no more secure as the 56 bit key space. It is possible to calculate DES with modern computing power. 3DES is more powerful than DES due to the three different keys, which is secure as its effective key length is 168 bits, but with two keys effective key length reduces to 112 bits which is less secure [21]. 3DES takes three times CPU power than DES which lowers its performance. AES outperforms 3DES in both in software and hardware. It uses 128-bit fixed length blocks and works with 128,192 and 256 bit keys. It shows the superiority of AES over DES and 3DES [21].

According to different studies on encryption algorithms found that AES algorithms needs less encryption time than RSA consumes the longest encryption time. Also decryption of AES algorithms is better than other algorithms. Using simulation results is evaluated that AES is much better than DES and RSA [22].

According to the comparative analysis of AES and DES security algorithms and found that different machines take different times for encryption/decryption of same algorithms over same data packet. Results showed that AES more secure as compare to DES [25].

Different simulation experiments shows that Blowfish has better performance than other commonly used encryption algorithms. Because of the more processing power AES showed poor performance compared to other algorithms. It shows also that AES consumes more resources when data block size is relatively big. In addition, the experiments proved that 3DES requires always more time than

DES because of its triple phase encryption characteristic. DES and 3DES are known to have worm holes in their security system. Blowfish and AES do not have any worm hole so far [26].

On the basis of detailed study of different symmetric key block ciphers discussed above, an attempt is made to critically analyzes them. DES is one of the encryption algorithms with some weak points. It uses same algorithm for encryption and decryption but the order of the keys is reversed in latter process. With rapid increase in processing speed of CPU and other advances in computing the key space of 56 bit key is considered no more secure from brute force attack.[26]

Algorithm type	Key size	Speed	Block size	Security level
DES	56 Bits	Slow, but speed depends on key	64 bits	Less secure
3DES	112/168 Bits	Very slow	64 bits	Moderately security
AES	128,192,256 Bits	Fast, but speed depends on key	128 bits	Secure
Blowfish	32 – 448 Bits	Fast	64 bits	Believed to be most secured
RSA	1024 Bits - and more	Fast, but speed depends on key	86 bytes	Secure

Figure 4. Comparison of some cryptography algorithms

CONCLUSION

In this paper, we have analyzed DES, 3DES, AES, Blowfish and RSA encryption algorithms. We have found that each algorithm has its own benefits according to different parameters. From the work completed in this paper it is observed that that the strength of the each encryption algorithm depends upon the key management, type of cryptography, number of keys, number of bits used in a key. Longer the key length and data length more will be the power consumption. For security view it is recommended to use short data sequence and key lengths. Since Blowfish has not any known security weak points so far, this makes it an excellent encryption algorithm to be considered as a standard encryption algorithm. AES showed poor performance results compared to other algorithms since it requires more processing power.

According required implementation memory DES and AES require medium size of memory. Also, Blowfish has smallest memory size. RSA consumes more time for encryption and decryption compared to others. Blowfish consumes the least time than other above described algorithms. Blowfish is efficient in software, at least on some software platforms. Evaluating DES, 3DES, AES, Blowfish and RSA we can conclude that Blowfish is strongest against guessing attacks. According different literature survey AES requires highest number of bits to be encoded optimally an encrypted data and DES requires least number of bits to be

encoded optimally. Evaluating by time and memory Blowfish is the best than others. According cryptographic strength AES is the strongest algorithm.

The present study provides critical evaluation of these algorithms. For future work can be done comparative or performance analysis with their different parameters to outline the strengths and weaknesses of various algorithms. Nowadays hybrid encryption scheme (symmetric + asymmetric) can have more security level and can ensure more strength, against of any third party attacks.

REFERENCES

1. Applied Cryptography, Second Edition: Protocols, Algorthms, and Source. Bruce Schneier, ISBN: 0471128457, 01.01.1996
2. Introduction to Cryptography, Second Edition, Johhanes A. Buhman, 2000
3. An Introduction to Modern Cryptosystems, Andrew Zwicke, 2003
4. Physical Security of Cryptographic Algorithm Implementations, Ilya KIZHVATOV, L'UNIVERSITÉ DU LUXEMBOURG, 2009
5. Handbook of Applied Cryptography, Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanston, Massachusetts Institute of Technology, June 1996
6. Diffie, Whitfield; Hellman, Martin (November 1976). "New Directions in Cryptography" (PDF). IEEE Transactions on Information Theory.
7. Cryptographic algorithms and reconfigurable hardware, A Brief Introduction to Modern Cryptography, F. Rodriguez-Henriquez, Diaz Perez, 2007
8. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы и исходные тексты на языке С, 2-е изд . – М.: Вильямс, 2003.
9. Введение в криптографию, Под редакцией В. В. Ященко, Издание четвертое, 2012
10. Основы криптологии, Д. Ананичева, И. Кориакова, 2006
11. КРИПТОАНАЛИЗ КЛАССИЧЕСКИХ ШИФРОВ, О. Н. ЖДАНОВ, И. А. КУДЕНКОВА , 2008
12. Баричев С. В. Криптография без секретов. – М.: Наука, 1998.
13. A. Sterbenz and P. Lipp, "Performance of the {AES} Candidate Algorithms in {Java}," Third {Advanced Encryption Stand. Candidate Conf. April 13--14, 2000, New York, NY, USA, pp. 161–168, 2000.
14. D. Coppersmith, "The data encryption standard (DES) and its strength against attacks", IBM Journal, Research Develop., vol. 38, no. 3, (1994), pp. 243 -250.
15. Introduction to Modern Cryptography by Phillip Rogaway and Mihir Bellare, 2005
16. D. Elminaam, "Performance evaluation of symmetric encryption algorithms," Int. J. Comput. Networks, vol. 8, no. 12, pp. 280–286,2008.
17. Ростовцев А. Г., Михайлова Н. В. Методы криптоанализа классических шифров . – М.: Наука, 1995.
18. Криптология – наука о тайнописи //Компьютерное обозрение. –1999.
19. Мао В. Современная криптография: Теория и практика — М.: Вильямс, 2005
20. Ященко В. В. Введение в криптографию. СПб.: Питер, 2001.

21. Hamdan O. Alanazi, B. B. Zaidan, A. A. Zaidan, Hamid A. Jalab, M. Shabbir and Y. Al-Nabhani, “New Comparative Study Between DES, 3DES and AES within Nine factors”, Journal of Computing, Volume, 2, Issue 3, March 2010, pp. 152-157.
22. Dr. Prerna Mahajan and Abhishek Sachdeva, “ A study of Encryption Algorithms AES, DES and RSA for Security”, Global Journal of Computer Science and Technology Network, Web & Security, Volume 13 Issue 15 Version 1.0 Year 2013, pp. 15-22.
23. Deepak Kumar Dakate and Pawan Dubey, “Performance comparison of Symmetric Data Encryption Techniques”, International Journal of Advanced Research in Computer Engineering and Technology, Volume 3, No. 8, August 2012, pp. 163-166.
24. Abdel-Karim Al Tamimi, “Performance Analysis of Data Encryption Algorithms.”
25. Sumitra, “Comparative Analysis of AES and DES security Algorithms”, International Journal of Scientific and Research Publications, Volume 3, Issue 1, January 2013, pp. 1-5.
26. Ayushi, 2010,A Symmetric Key Cryptographic Algorithm, International Journal of Computer Applications (0975 - 8887) Volume 1. No. 15, 2010
27. "Quantum cryptography: An emerging technology in network security". - Sharbaf, M.S. IEEE International Conference on Technologies for Homeland Security . 2011
28. The official Advanced Encryption Standard" (PDF). Computer Security Resource Center. National Institute of Standards and Technology. Retrieved 26 March 2015.
29. "The Digital Millennium Copyright Act of 1998" (PDF). United States Copyright Office. Retrieved 26 March 2015.