IJCRT.ORG ISSN : 2320-2882
INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

Systematization of Knowledge: Cybersecurity Techniques from a Military Intelligence Perspective

Eduard Simion
Faculty of History, International Relations and Political Science University of Oradea, Romania
1 Universitatii Street, Oradea, 410087, Romania

Abstract
In this paper we systematize the knowledge on several cybersecurity use cases. We commence with a historical prelude, having in mind to identify military-relevant use cases. We continue with a review of several known and publicly disclosed cybersecurity techniques. Specifically, we go over classical defense options against distributed denial of service attacks and cover anonymization schemes. Perhaps more importantly, we consider their applications in the field of cyberdefense as well as cyberwarfare.
Keywords: cybersecurity, cyberdefense, cyberwarfare.

1 Introduction

Intelligence agencies represent a foundational landmark, lying at the basis of all modern states. Historically, their main objective was to target economically- or military-relevant data, on friends or foes, an activity casually denoted through the terms intelligence gathering or espionage. The other main objective is the protection of state's critical structures, civil or military, from interfering with external parties (counter-espionage). In a broad sense, we place the latter term, in a larger set of activities, denoted counter-intelligence, which include gathering economic or social relevant intelligence on the internal ecosystem of a country, along with halting counter-espionage activities.
In our work we make use of the taxonomy defined above. We commence with a historical per- spective over the fascinating field of intelligence, by looking over the classical modus operandi of some publicly-disclosed cases. Then, we analyse the main challenges to be addressed given the current means, with an emphasis on cyberattacks and cyberdefense strategies.

1.1 Intelligence Gathering - Defining the Targets

In this section we provide a historical overview of intelligence activities. Our purpose is to identify the kind of informations that are usually targeted. We interleave historic cases encompassing both civilian and military operations. In Section 2 we will come back to the easily understandable example and survey over methods used to achieve such goals.

The very first references to intelligence are as old as the Biblical Book of Judges: Joshua, the son of Navi, is sent to observe the activities inside the fortress of Jericho. Jericho was the cornerstone in conquering the Land of Israel. Therefore, knowing the positioning of your enemies' troops and their specific vulnerabilities, before a battle, proves to be a key ingredient in the success of any military operation.

1.1.1 Conquering Jericho: From Joshua Navi to T.E. Lawrence

Thomas E. Lawrence, a colonel of the British Army, made several remarkable contributions to the outcome of the First World War in its Middle Eastern theatre.

Lawrence was a historian with extensive interest in Middle East. He spent the years before the outbreak of World War I in expeditions investigating Hittite settlements at Carchemish, near Eu- phrates, in modern day Turkey and Iraq. Most importantly, he became interested in the culture of the people and learnt their language, becoming an Arabist. In 1914, together with other members of his expedition, Lawrence explored the Sinai peninsula, which bordered the Ottoman Empire at that time. During the outbreak of the Great War, Lawrence was posted in London, as an employee of War's Office Map Department.

He was soon posted to Cairo, and by the end of 1914 was already part of the Military Intelligence Bureau of the British Army in Cairo. For completeness, we mention that the Ottoman Empire was an ally of Germany and Austro-Hungary, while the British Army had opened two fronts against the Ottomans: in Egypt and in the Arab peninsula. Lawrence's daily activities included map drawing, prisoner interrogation or providing analysis on the data gathered by the Army.

The genius of Lawrence, that brought his nickname of Lawrence of Arabia, consists of his series of brilliant observations: the tribes in the Arabian peninsula, under Hussein ibn Ali, who ruled over Mecca, were willing to start a generalized revolt against the Turkish Army. Prince Faisal, a son of Hussein, was ruling his troops around Medina. Lawrence joined Faisal's army as an adviser. With enough determination, he started the massive revolt of the Arabs against the Ottomans. His actions were rapidly noted by General Allenby, the commander of the Egyptian Expeditionary Force, who gave Lawrence some freedom in advising Faisal.

Most importantly, having the chance to advise the correct decision-makers is a fundamental element that must be central for every military-intelligence strategy. The story behind the life of T.E. Lawrence set importinates co-ordinates to be followed: learning the adoptive country's language, knowing in detail the peculiarities of the culture, and cultivating strong inter-personal relations.

1.1.2 Looking through Your Enemy's Glasses: Penkovsky, Polyakov, Gordievski

Three of the most notorious cases of espionage during the XXth century were those of Oleg Penkovsky, Dmitry Polyakov and Oleg Gordievsky, all members of the Russian intelligence agencies.

Oleg Penkovsky was the son of a Russian officer who served in Czar's army and fought against the Bolshevik party, during the Civil War. This fact had a significant impact on Penkovsky's future career. Penkovsky enrolled into a military artillery school and joined the Red Army in their 1940 campaign against Finland. His career took an ascending trend after his marriage: he enrolled the Mikhail Frunze Military Academy, assigned to GRU and went to intelligence training. His assignment to Turkey was very prolific. However, he has been denied promotion (repeatedly), a fact that was instrumental in his decision to turn the sides. Penkovsky contacted the British intelligence while posted in London, and has been in contact with them from Moscow. He offered extremely precious informations on Soviet's plans for sending missiles to Cuba, perhaps preventing a nuclear war.

Dmitry Polyakov raised to the rank of Major General in Soviet Military Intelligence. He was probably the highest placed asset the US intelligence community ever had in the Soviet intelligence. Polyakov was disillusioned from the Communist society after his repeated requests for getting Western medical support for his son were blocked by the Soviet apparatus. His main contributions were on the degrading relations between the Soviet and Chinese states, a fact that has been exploited during the years of Richard Nixon.

Oleg Gordievsky, the son of an NKVD officer, was one of the top British spies in the heart of Soviet intelligence. Gordievsky was a brilliant student, mastering German, and learning Norwegian, Danish and Swedish. Later, he learned English. During his first assignment, Gordievsky was posted to Copenhagen. He defected for ideological reasons, being disillusioned by the armed intervention during the Prague Spring. Since from 1982, he had been posted to London, where he became station chief. He was recalled to Moscow in 1985, being under suspicion for treason. Gordievsky escaped with the help of the British Secret Intelligence Service, the same year.

1.2 Science and Technology Intelligence

The second group of intelligence-gathering activities can be centred around gathering scientific and technological intelligence. As a central remark, this kind of espionage was (and still is) more common to emerging economies that do not have the financial power to invest in research and development. Thus, the rely on stealing technology from well established Western corporations.

1.2.1 Ion Mihai Pacepa - Stealing Technological Secrets

Ion Mihai Pacepa was probably the highest-ranked intelligence asset the US intelligence community ever managed to get during the Cold War. Pacepa was born in 1928, as the son of a car dealer having Communist sympathy. He graduated from the Polytechnic Institute of Bucharest in 1950, and got a job in the Securitate. Several versions exists on how Pacepa acquired the position, including rumours of his father's friendship with the Soviet leadership in Securitate during the 50s.

His formation, as a chemical engineer, as well as his knowledge of German, recommended him for foreign operations. As a DIE (the foreign branch within Securitate) operative, Pacepa was firstly assigned as a commercial attache in Franfkurt, West Germany. He was responsible to the scientific and technological espionage. This was realized, for instance, by placing Romanian engineers to Western companies working in sensitive areas. Some of his main contributions were in strong relationship with the rise of the petrochemical industry in Communist Romania. Pacepa rose through the ranks, and by 1968 was a top adviser of the newly installed Romanian leader, Nicolae Ceausescu.

He held the position of deputy chief of DIE, focusing on S&T espionage. The main positions was completed by Nicolae Doicaru (the director) and Gheorghe Marcu (external commerce). His relation with the Ceausescu couple was very close, being a main organizer of Ceausescu's visits in the United States and the United Kingdom. By 1978, Pacepa should have taken the newly created role of Director of Presidential House. He became disillusioned with the given tasks (including the elimination of hostile Romanians from diaspora) and decided to flee. He went to the US Embassy in Bonn, was flown to the States, and lived there under a covered identity by the end of his life.

Pacepa published several books, including [5, 6], which present his points of view over the Com- munist Romania, the Ceausescu couple, and the Romanian methods for performing espionage. his top position gave the Western counter-intelligence the chance to find the identities of many Romanian operatives.

Pacepa's case, as the head of an intelligence-gathering branch targeting scientific and technological informations

1.2.2 Robert Hanson - Stay Alert, Stay Alive, Stay Anonymous

Robert "Bob" Hanson was one of the highest placed Soviet/Russian spies inside the US community service (working within FBI). Together with Aldrich Almes, he produced a high damage, by exposing Polyakov for example. Hanson was born in Chicago, as the son of a law-enforcement officer, majored in Chemistry, and learned Russian as a foreign language. Later, he earned an MBA in accounting and information systems.

His career with the Bureau started in 1976, initially in Gary, Indiana. He was perceived as a highly intelligent agent, but extremely introverted. He moved to Washington from 1983, and was given clearance for highly sensitive informations within he Soviet Analytical Team. In 1987, he was transferred back to New York.

Hansen betrayed mainly for financial reasons, by selling documents to the Soviets. He was rewarded for his actions. After a period of silence, he approached again the Russian intelligence, by contacting a known, but unsupervised agent operating within the United States. He provided names of three Russians working for the US (two of them were executed), and requested large amounts of money for informations. Over his career, he was paid more than 600000 USD. Hansen passed intelligence on US nuclear weaponry, satellite communications.

The interesting aspect in Hansen's story was his willing to remain *anonymous*. He never provided his Soviet handlers his real name, which prolonged greatly his activities.

1.3 Contributions

Our contributions are threefold. Firstly, we consider the problem of defending systems against denial of service attacks. Such a problem is of paramount importance in nowadays societies, for both civilian and military uses. In section 3, we look into an approach that use time lock puzzles. We take a deeper look into the problem of secret generation for time lock puzzles; our work avoids the classical approach of storing secrets in a database and checking their validity; we propose a novel solution to generate puzzle secrets using symmetric key encryption.
Secondly, in section 4, we tackle a specific military problem, of anonymizing and signing case reports. The problem is of major, real importance for many intelligence branches. In our problem description, *n* case officers are sending reports to their direct superior. We would like reports to disclose essential details exclusively to their direct superior, and to prevent the direct superior (or indirect ones) to figure out the author of the report.
Thirdly, section 5 analyses the usage of machine learning techniques for early warning in military conflicts. This is particularly important, especially when massive amounts of data are available in short time to analytical teams consisting of few people.

2 Background on cybersecurity

Cybersecurity has become a very broad term since the Internet's Big Bang. As computer networks expanded more and more, interconnecting large numbers of computers, individuals were granted access to data hosted on someone's else machine[1]. From a non-technical perspective, several questions arise:
a) what happens when I download a page? Can it interact with data displayed on a different page?
b) What happens when I deploy my data to a sever? Who has access to it? c) Shall I run a program downloaded from an untrusted source? d) Who can read my emails? How can I protect them? below, we try to classify the most common cybersecurity threats and review some of them relevant for military applications.

2.1 Definitions

Craigen *et al.* [1] define cybersecurity as a cluster of methods that helps to protect, defend or attack system storing or exchanging informations.
In a broad sense, we can define cybersecurity having in mind the adversarial models. Usually, the attacker would like to exploit:

- Data-privacy: the adversary would like to get access to data that are stored on victim's machine. When obtaining the entire set of informations is infeasible, even partial pieces may be relevant. More sophisticated attack try to learn informations by *distinguishing* between the data representation[2] in a military context, without having to consider a full-scale attack.

- Data-integrity: in such a model, the adversary attempts to exploit the malleability of data, by inserting, or removing important pieces [3].

- Data authenticity: the attacker mimics a benign entity, by sending messages in his name. In hierarchical institutions, this constitutes an extremely powerful way of propagating disinformation on the field.

A toy, demonstrative example of attacking the authenticity of the information displayed, on the client side, works as follows: a simple script, such as TamperMonkey can be installed on victims's bowser to alter the contents of a webpage once the browser loads it, most of the time disinforming the genuine user.

[1]In modern era we commonly speak about clusters of machines denoted as "clouds".

[2]For instance, if the adversary can distinguish between two emails that read: *Attack now!* or *Attack later.*, it may be enough

[3]If the message reads: *Attack now at Aqaba.*, it can be changed modified to *Attack now at Jericho.* .

```
// ==UserScript==
// @name         New      Userscript
// @namespace    http://tampermonkey.net/
// @version      0.1
// @description  Demo Script
// @author       Demonstrator
// @match        https://www.favurite.website/*
// @grant        none
// ==/UserScript==

(function() {
    'use strict';
    alert('Your account was hacked');
})();
```

- Network issues: network issues are the common ways of allowing attackers to access codes in nowadays systems. In the current architecture of open source operating systems, applications run at specific ports. Leaving these ports unprotected may allow access by adversaries. More damage is made when specific applications run at known ports, and the adversary benefits exploits known vulnerabilities which were not yet patched.

Another example of networking issues, that has been recently discovered is linked to somehow surprisingly, a logging library [3], which simply logs app information that help debugging in classical software engineering tasks.

- Malware: is probably one of the most used type of attack. It includes sending trojan viruses, spywares or bot executables. It is also one of the broadest classes of attacks, as it includes a large plethora of viruses which are easy to replicate. The common defense system is the usage of antivirus programs. The main principle behind an antivirus is relatively simple: a large set of database of virus signatures is stored, while the file system is scanned and checked if it matches any of the known signatures. Once new malware occurs and get detected, the database increases.

Common scenarios of deploying malware is through USB devices, which may be given as gifts for instance, although their distributed may have pre-installed unwanted software on it[4].

2.2 Common Cybersecurity Topics

An evolving number of topics can be linked to cybersecurity techniques: among these, we include reverse-engineering of programs, cryptanalysis techniques, networking issues, data-mining or machine learning. Out of them, we consider four for in-depth scrutiny, given their increased incidence in our study.

2.2.1 Denial of service attacks

Denial Of Service (DOS) attacks occur when a publicly available resource becomes “bombarded” by numerous requests, meant to make it inaccessible. Usually, this is done either by flooding the target with traffic or by exploiting software bugs that crash the application.

Most of the time, DOS attacks are mounted against web servers. The preferred targets include web-banking services, military/governmental infrastructure or social media networks. The requests that are made are classical HTTP requests, while the web server (e.g. Appache Tomcat, Nginx, etc) cannot serve all requests. The problem becomes telling apart genuine users from attackers.

A particularly concerning type of DOS attack, is a so-called Distributed Denial Of Service Attack (DDOS): in such a case, the flooding requests are generated by a very large group of users which may be running, for example, a designated app. The attack can be made possible by specific code fragments asking to send requests to the APIs put forward by the victim’s website[5]. The losses incurred by

[4]For instance, spyware may record the keywords that are pressed, including passwords.

[5]Consider a free app that is measuring your pulse rate, which can be downloaded from Google’s Play Store. If the app’s API allows to trigger HTTP requests to external IPs when specified via a “remote maintenance” API , then such

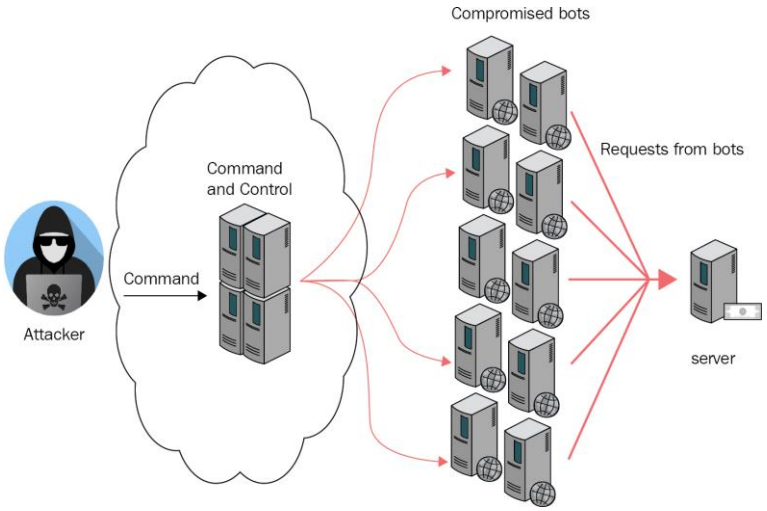

Figure 1: A denial of service attack ([2]) includes the following party. The attacker controlling a set of compromised assets. At a given point in time, thy execute simultanously requests on a server.

flooding attacks translate into very large sums of money, especially for banking service, where millions of clients may be affected for performing daily transactions. The modern solution to DDOS are the firewalls. According to Palo Alto security guidelines[4], DOS flooding attacks can be usually classified as falling within one of the three categories:

- Buffer overflow attacks - are realized by flooding a server with traffic, hoping the victim will fail in handling so many requests.
- SYN flood - a connection request is sent, but the handshake is never completed. this is continued until the pool of available ports is completely saturated.
- ICMP flood - uses misconfigured network devices (routers, switches) in order to "spoof" packets that ping the entire network. The traffic is going to increase doing the network.

The second way of realizing Denial of Service attacks is by exploiting the software vulnerabilities of the software. This is a very broad area, and is very much programming language dependent. For instance, attackers may exploit webpages that do not "digest" users' passwords into short length, ran- dom looking strings. In many cases, if the attacker gains access into the system and if he has admin rights (this can happens for instance by using weak admin passwords), then a ransomware attack can be produced.

Ransomware attacks happens as follows: (1) the attacker generates a pair of public/private encryption keys: PublicKey, PrivateKey. (2) a symmetric encryption key having a length of k bits is generated:

$$\text{SymmetricKey} \leftarrow \{0,1\}^k$$

(3) the SymmetricKey is encrypted under PublicKey.

Ciphertext = RSA.Encrypt(PublicKey, SymmetricKey)

(4) the ciphertext that results after the encryption procedure run, is sent back to the attacker; (5) the attacker encrypts the victims' file system:

VictimsCiphertext = AES.Encrypt(SymmetricKey, FileSystem)

An amount of money is requested from the victim to recover SymmetricKey. The amount is requested to be paid in anonymous cryptocurrencies. Ideally, such payment systems [7, 8], should le be anonymous.

an app may be used to call a specific web-server API, when specified by the designers. This model can represent a successful DDOS-source, especially when such an app passes the bar of several hundreds of thousands of users.

2.2.2 Anonymity

From a cryptographic point of view, the notion of anonymity asks anyone to try to distinguish if a ciphertext has been generated under a key or another. If the ciphertexts produced by encryption schemes are (pseudo)random, then an encryption scheme fulfils anonymity. In this work, we assume we work with standard symmetric cyphers (AES) and (padded) public-key schemes (RSA), as specified by the newest TLS standards.
When speaking about anonymity in ransomware attacks, we usually refer to the notion of *untrace- ability*. From this point of view, the attacker would like to:

1. hide its connection details (the IP real, the MAC address, the real Internet service provider, the host country, the organization he belongs to),

2. make sure its account identifier (usually a public key) cannot be linked to his other metadata.

Anonymity involves knowledge on public key encryption and signature scheme As shown in Section 3, anonymity may be used in military intelligence.

2.2.3 Data privacy through encryption

Privacy ensures that information can be sent through a large variety of meaning, and only the sender an receiver are aware of the actual content of the information, under the assumption that it is received intact.
Encryption is the prime method for realizing encryption. We have already discussed the impor- tance of encryption, while taking about ransomware attacks. At least from an attacker's point of view, is a crucial topic. We see here it matters also for a defender. Encryption techniques can be classified as public key or secret key.

Private-key encryption, is the classical way of encrypting information. A secure cryptographic key is pre-distributed between two users. The encryption and decryption has the following syntax:

$\text{Ciphertext} = \text{Encrypt}(\text{sk}, x)$, $x = \text{Decrypt}(\text{sk}, \text{Ciphertext})$.

A main problem is how many keys are used, and how often are these going to be changed. AES is the gold standard used in private-key encryptions schemes.

Public-key encryption can be used to communicate without pre-sharing a cryptographic key, but having a public one. RSA is one of gold standards. The syntax of public key encryption is as follows:

$\text{Ciphertext} = \text{Encrypt}(\text{pk}, x)$, $x = \text{Decrypt}(\text{sk}, \text{Ciphertext})$.

2.3 Cybersecurity in a modern military-reconnaissance context

Modern military interactions make extensive use of intelligence on adversaries position of troops, the size of the military units involved, the type of military technique and its state, the ability to react in short amount of times, the communication between command and field units. Reconnaissance collects intelligence on such sources. Several ways of performing it are enumerated below:

1. Satellite systems: an extremely expensive mean for providing imagery of large *ground* areas, which makes it usable by few operators. The advantages are covering large areas, as well as protection against attacks able to be mounted by classical adversaries.

2. Drones: an affordable mean for limited-size budget armed forces. Drones are easy to operate, can provide high quality imagery of relatively large areas, but are prone to identification and direct hits by adversaries.

3. Reconnaissance patrols: can be deployed as ground forces or aerial. The main disadvantage consists in exposing valuable human resources in enemy territory. However, the information they may provide may be qualitatively superior to what satellites or drones may obtain.

Storing and transmitting informations between the source that gathers data and the receiver(s) remains a pressing problem. Cryptography ensures confidentiality. A symmetric-key cipher, such as the Advanced Encryption Standard AES, allows to process blocks of 128 bits as input. Its main advantage is fast speed of processing the data. However, to increase live streams of data, AES must need to work in a mode of operation, such as CBC mode.

There are several practically-relevant problems which occur: can we distribute information to multiple receivers? For example, can a drone send, in real time high definition images to a platoon operating covertly? A more pressing question is What happens if one receiver looses his communication device: is the communication compromised? On the analytic side, a major problem may be the large amounts of information and the inability to process them.

3 Techniques for defending against cyberattacks

3.1 CAPTCHA codes against denial of service attacks

CAPTCHA codes became a very popular solution to prevent (distributed) denial of service at- tacks. Themselves, CAPTCHA codes (and their derivatives), are human solvable puzzles, involving a sequence of "malformed" characters that are to be easily recognized by humans. Oher variations (by Google services) involve images containing specific objects, and ask humans to distinguish images containing such objects from those that do not. Other CAPTCHA-related techniques involves objects that are to be moved, or rotated, to match certain position or puzzle.

In short, CAPTCHAs are human solvable puzzles. The interesting aspect is that CAPTCHAs do not solve the (D)DOS threat. The important aspect is they allow to distinguish human users from automatically generated requests. Still, the webpage that loads the CAPTCHA may be accessible for large number of requests in short time. However, this is accepted as long as no significant damage on the persistent data (stored in databases) or on the backend layer is done.

CAPTCHAs become irrelevant when a large amount of humans decide to use an application: for instance, a public TV announcement about a website may trigger a genuine increased traffic. CAPTCHAs are useless in such situations.

3.1.1 New Perspective: Time-Lock Puzzles Against Denial of Service Attacks

A Time Lock Puzzle (abbreviate TLP) is a method for disclosing data in the future. For example, a piece of data that benefits for classification until 2100, can be safely "encoded" with the help of a time lock puzzle. An encryption key can be used to encrypt data, while the decryption key (the piece of information that allows to recover it), can be hidden under a time lock puzzle for a long period of time.

We present here the introductory time lock puzzles, presented in the work of Ronald Rivest, Adi Shamir and David Wagner[9]. Consider two safe primes p and q and an RSA modulus

$$N = p \cdot q,$$

where the factorization of N is not disclosed. When we say that a prime number p is safe, we mean they were generated as follows:

$$p = 2 \cdot p' + 1,$$

where p' stands for an arbitrary large prime number. The safe prime q is obtained in the same way. The hard problem behind the RSW time lock puzzle says that the calculation of

$$g^{2^T} \mod N,$$

without knowing the order of the underlying group, is a difficult task for all attacker. It is presumed that most best way of performing this calculation is through an iteration over T modular and repeated squarings operations. To standardize the problem, it is suggested to set:

$$T = 2^t.$$

Thus, it easy to observe that T acts as a parameter that controls the difficulty of the problem. On the other hand, if the factorization of N is known (as it is the case) of the puzzle generator, it is easy to calculate:

$$g^{2^T \bmod \varphi(N)} \bmod N,$$

and obtain the same result.

To prevent denial of service attacks, the server is supposed to generate some secret s, which is to be encoded under a TLP as follows:

$$puzzle = s \cdot (g^{2^t} \bmod N)$$

Once $g^{2^t} \bmod N$ is retrieved by a faithful follower of the login protocol, s can be extracted and returned to the server. An implementation of the protocol is provided by the following code-snippet (written Crystal-lang).

```
class RSW

  def initialize
    @p =1397814462664361842482169990350776301470104271800016537222821107484278
    836644287859924799794591133092831626520423403173987689364558006105852721424530
    105389654530297792257102364875329531757098530540471483418384839217300259600233
    269517132831100352926284178759515325533671235264537388684163427298539178763344
    59262673
    @q = 1572216746562380675828563105973396626086920677969066020832312879965632 4
    977774375433060858851784511648983740587893064426653194099393954841102553899 1
    197596216803053901580138007071597089292024343820093619972285300301850726245 3
    754168374663825482727103170167677486498055883189180335620177141332576899987 5
    3808221187
    @modulus = 2197667306788005308350005469427233452812994860712781335183840595 81
    3950281645912484908849019524350873992003921774863269879852747094680997663671 2
    4572270247540302536382518460552090602633874135823593639281289972213515701364 0
    4807700589866870703604835644500564172053946122422841983109386408346542794330 5
    2165387858739215705117753099656403279388909694964182721226684840929946404495 0
    1019689941784456473541684584507030557410355210813656252234465209008651429345 5
    1560213448911754529522482791102193287137545853797806197158140036590557966806 4
    6868241064375822449532982155843646402782474805407780430453068615111254283581 2
    962716852851
  end

  def rsw_generate
    g = Random.new.next_int
    return g % @ modulus
  end

  def rsw_solve(g : BigInt, time : Int32)
    i = 0
    solution = g
    while i < 2^ time
      solution = ( solution ^ 2) % @modulus
      i = i + 1
    end
    return solution
  end

  def rsw_verify(challenge : BigInt, solution : BigInt)
    if pow(g, 2 ^ t % (@p - 1)*(@q - 1)) % @modulus == solution
      return true
    else
      return false
    end
  end
```

3.1.2 Batching time lock puzzles

In this part, we contribute and propose a notion of batchable TLPs, which should increase the chances to defend against denial of service attacks. Imagine that the login webpage of a institution is queried. We want to provide each user with a time lock puzzle that encapsulates a secret.

The major topic that we digress in this work what happens when the server is bombarded with a large number of "faked" secrets, after a period of T core-units passed. There is a need for one efficient way to verify if these TLP-expected secrets are genuinely generated. A simple way for the server is to store the secrets. The problem occurs when the same SQL database offers both reading and writing capabilities. This may become prohibitive if hundreds of thousands/ or even millions of requests are made: the pression on the database increases, as database is usually locked for writing and committing new data, and unlocked to be read. To change this, we propose the following strategy.

Proposal 1. (Time Lock Puzzles against DDOS) For a time lock puzzle construction TLP and a secret key K stored by server, generate the secret s as follows:

$$s_0 \leftarrow \{0,1\}^{128}$$
$$s_1 = AES(K, s_0)$$

Then generate the time lock puzzle as:

$$puzzle = TLP.GenerateP uzzle(params, s_0 1 s_1) .$$

The verification (server-side) recovers $s_0 1 s_1$. *Then check if:*

$$s_1 = AES(K, s_0) .$$

We motivate here why the construction fulfils its needs. In case of a denial of service attack, the server does not have to rely on the database layer in order for checking if the secrets are valid. All is needed is the AES key K. More importantly, Intel processors come with an AES instruction, which makes the evaluation of AES extremely fast.

4 Anonymity for writing reports

A significant aspect of any intelligence gathering operation is providing a descriptive report on the tasks that have been accomplished and on the informations that has been gathered. Usually, this comes along with a signature (or other identifier) of the person who was assigned the case.

The story of Robert Hansen described by our introductory exposition is illustrative: sharing the names of sources, or even of their handlers amongst large organization may be fallacious: when someone has access to all these names and may be decided to "leave", the damage may be incredible.

To prevent such a scenario, we propose the usage of i) ring signatures in conjunction with ii) sensitive-field anonymization through encryption. We digress into our military setting: we assume that n case officers are part of a known group, all being known by their superior(s) (located in the central), all owning a pair of secret/public keys, and all knowing the keys of their colleagues. This distribution of public-keys in the group can be done by their superior. Thus, each officer knows the public key of the group member, although he must not know the identity of his colleagues.

Ring signatures are designed for such a scenario. This intriguing way of signing messages is presented section 4.2.

The other problem concerns the human ability of interpreting the messages and deducing who the author was based on their description. When one officer only is posted in one country, and the report describes that specific country, there is a smoking gun: everyone reading the report (the entitled superior or unauthorized superiors) may deduce the origin of the source. We already assume all reports are encrypted under an institutional key (otherwise the counter-intelligence's jobs of the host country is simple).

4.1 Anonymization techniques

As exlained above, it is often the case that descriptive reports (contains only text data) includes a sensitive set S of keywords. A comprehensive definition of what S can contain is hard to get, but we assume that:

$$S = \{\text{names, location, dates, descriptive nouns, units of measure}\} .$$

Assuming a report includes such keywords, and the report is addressed to X, identified by apublic key PK_X, we propose their replacement as follows:

1. encrypt each keyword w belonging to S, indenpendently, under PK_X, and obtain the ciphertext C_w.
2. replace w by C_w in report R and obtain an anonymized report R^l.

4.2 Ring signatures background

A ring-signature (RS) is a method to hide the real signer among a large group that can each actually produce a signature. They were introduced in [10] by Rnald Rivest, Adi Shamir and Yael Kalai and enhance on theproperties of group signatures[11]. Ring signature do not need a central authority to manage the key creation and revocation, as it is the case of group signatures. The requirement is that every group member knows the key of the other parties. Such ring signatures can be instantiated from the RSA cryptosystem and some other additional primitives.

We apply the method described in section 4.1, and obtain an anonymied report R^l

Signing: choose a symmetric key K as the hash of the report R^l to be signed,

$$K = Hash(R^l) .$$

Then choose a value $v \leftarrow \{0,1\}^b$. *Choose a random* x_i *for all the other* $n-1$ *ring members, and compute:*

$y_i = g_i(x_i)$.

Recover y_s *by solving the ring equation (that depend on K and* y_i *). Calculate* x_s *using the signer's private key as*

$$x_s = g_s^{-1}(y_s) .$$

The ring signature is set to be the tuple:

$$(P_1, \ldots, P_n, v, x_1, \ldots, x_n)$$

Verification: compute $K = Hash(R^l)$. *Apply the public key trapdoor to obtain*

$$y_i = g_i(x_i) .$$

Solve the ring equation and verify if its outcome is v.

Clearly, byt the security of the ring signature scheme, everyone can check the ring signature has been signed by one of the n members. The anonymized report R^l can reveal its sensitive information set of words S only to person (group) X.

5 Machine learning techniques for military intelligence

Machine learning techniques are extremely useful in processing massive amounts of data. Out of the tools that are available in this area, deep learning has a central role. The purpose of deep-learning is to "imitate" the way human think.

The importance of machine learning increased over the past years, as the Internet traffic exploded and automatized solutions were envisioned. We discuss the pros and cons of using machine learning in military operations.

5.1 Data analysis in aerial imaging

We evaluate the usage of aerial imaging that are obtained usually through satellites or drones. Earlier imaging has been done through spy planes. The history of employing them proves how risky it is, since the human piloted planes are subject to anti-aerial defense. However, since the occurrence of satellite systems or unmanned aerial vehicles, the risks decreased.

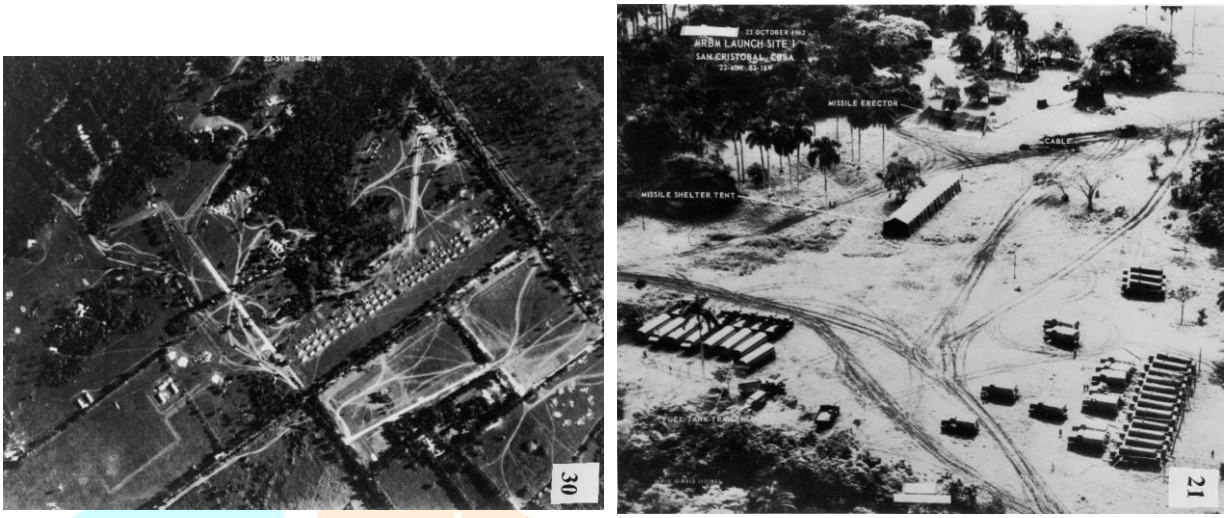

Figure 2: Two images [13] from a U2 plane presents the organization of a Soviet base in Cuba. While the right image presents an easily identifiable cluster of military equipments by humans, the left image makes it hard to identify such details to humans.

The first gain in using machine learning consists of identifying the military technique used. By comparing the ratio between vehicles to the scale of the map, precious informations can be dissemi- nated:

- what types of armoured fighting vehicles, tanks, howitzer are dislocated?
- what are the numbers of these military components

The most important thing is that recently, machine learning became better than human vision in distinguishing features for long range. Thus, it can be more reliable to employ machine learning in comparison to classic human analysis.

5.2 Early warning systems

The other aspect we consider in our work, is the ability of modern cyberwarfare techniques involving deep learning to be able to prevent large scale attacks by viewing movement of troops. We must mention, though, that early warning systems are already in place, usually monitoring the launch sites for ICBMs.
We foresee two main use cases of such early warning systems:

- preventing large scale invasion of states and the respective movement[6] of troops; detailed analysis may provide very good estimates on the size of troops involved.
- in combat zones, multiple drones can capture real time images from multiple corners of a city. Machine learning can be used to clusters, and further identify the vehicles they are using.

The usefulness of such scenarios are more actual then ever, in the context of countries being threatened by large scale invasions. As such invasions are largely terrestrial, machine learning can be successfully applied on top of images captured by satellites or drones.

[6]However, such actions can be observed equally by ground human sources.

Figure 3: Kabul drone imaging [14].

6 Conclusion

Our work identifies some major objectives of cybersecurity and cyberwarfare, based on historical examples. We make an overview of what cybersecurity means through its modern and rather lax definitions. Then, we consider several use cases of cybersecurity, with an emphasis of military systems that need to be always stable and always available. We also consider a pressing problem of sending reports from a case officer to its superior, and how to prove the report has been signed by someone without disclosing it.
The open problems are mostly related to our third application: processing large amounts of data and extracting the relevant ones. We only consider a case study of images obtained by satellites, but similar mining techniques can be applied to massive text data.

References

[1] Craigen, Dan, Nadia Diakun-Thibault, and Randy Purse. *Defining cybersecurity*. Technology Innovation Management Review 4.10 (2014)

[2] [Online] Available: https://exploitszone.com/wp-content/uploads/2020/06/ddos-attack.png, Accesed on December 2021.

[3] [Online] Available: https://edition.cnn.com/2021/12/15/tech/log4j-vulnerability/index.html, Accesed on December 2021.

[4] What is a denial of service attack (DoS)? [Online] Available: https://www.paloaltonetworks.com/cyberpedia/what-is-a-denial-of-service-attack-dos December, 2021.

[5] Ion Mihai Pacepa. Red Horizons: Chronicles of a Communist Spy Chief, 1987. ISBN 0-89526-570-2

[6] Ion Mihai Pacepa The Kremlin Legacy, 1993

[7] Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." Decentralized Business Review (2008): 21260.

[8] Wood, Gavin. "Ethereum: A secure decentralised generalised transaction ledger." Ethereum project yellow paper 151.2014 (2014): 1-32.

[9] Rivest, Ronald L., Adi Shamir, and David A. Wagner. "Time-lock puzzles and timed-release crypto." (1996).

[10] Rivest, Ronald L., Adi Shamir, and Yael Tauman. "How to leak a secret." International Conference on the Theory and Application of Cryptology and Information Security. Springer, Berlin, Heidelberg, 2001.

[11] Chaum, David, and Eugène Van Heyst. "Group signatures." Workshop on the Theory and Application of of Cryptographic Techniques. Springer, Berlin, Heidelberg, 1991.

[12] [Online]. Available: www.hefce.ac.uk/pubs/hefce/2009/09 39/, Accesed on December 2021.

[13] [Online]. Available: https://nsarchive2.gwu.edu/nsa/cuba_mis_cri/ , Accessed on December 2021.

[14] [Online]. Available: https://akm-img-a-in.tosshub.com/indiatoday/images/bodyeditor/202108/6-1200x1587.png?dHpQxAQqurncpgqw86_j.Tuw5zeIul7g , Accessed on December 2021.