Abstract:
Cybernetic Protectors is to provide a secure way of communication and transferring evidences in Secret Intelligence Agency of defence system which always uses undercover agents to solve complex cases and dismantle criminal organizations. We are conceptualizing this software as a solution so that Secret Intelligence Agencies and their agents can communicate through this Software for the exchange of evidences in a secure way, and maintain the details of main officer.

Keywords—Secret Intelligence Agency, Security, Face Recognition, Digital Signature.

I. INTRODUCTION

The Secret Intelligence Agency is the nations first line of defence. It accomplishes what others cannot accomplish and go where others cannot go. It carries out the mission by collecting information that reveals the plans, intentions and capabilities of the adversaries and provides the basis for decision and action.

The Cybernetics Protector is software which allows a security agency to handle various confidential missions in a secured way.

The Cybernetics Protector software is concerned with the security of the country and thus proper care has to be taken that confidential data from within the database is not leaked out.

Every country requires a Secret Agency who undertakes cases which are a threat to the national security. These agencies operate with the help of undercover agents who help solve these cases. Since these cases deal with the nations security, the communication and data transfer between the agents and higher authorities need to be protected. Hence developing such a system is necessary to help these agencies operate in a secret and secured way. The system will be used by a set of five different users. These users are Defence Ministry, Chief, Agents, employees and Citizens of the country.

II. SYSTEM FUNCTIONALITY

Figure 1 explains overall functionality of the system along with different set of users involved. In the information systems development field, requirements form the foundation for the rest of the software development process. Since building a high-quality requirements specification is essential to ensure that the product satisfies the users.

![Figure 1 Cybernetics Protector Users](image)

**The Defense Ministry** - The Defense Ministry assigns cases to the Secret Agency and allocates resources to it. It should be able to receive reports regarding the cases.

**The Security Chief** - The Chief of the Secret Agency has the highest powers. He can administer the agents, assign cases and resources. Also he has right to view the database.

**The Agent** - The undercover agent can send the evidence and data collected in an encrypted fashion so that the data cannot be intercepted.

**Citizen** - A citizen has the lowest access rights. A citizen can only view the success stories of the agency and chat with the officials.

The functions of these different users shown in Figure 1 are as listed here,
1Agent Manipulation
This feature is provided to the Chief of Security. The Chief will be able to Add/Delete/Edit Agent Records.

2. Agent Appointment
This feature is provided to the Chief of Security. The chief will appoint an agent for the case.

3. Secure sending and retrieval of data
This feature is provided to the Chief of Security, Agent and the Defense Ministry. This feature basically enhances the security of the software.

4. Access of Data Logs
This feature is provided to the Chief of Security. This feature enables him to analyze the data logs.

5. View Case Details
This feature is provided to the Agent. The agent will receive the entire case details from the Chief of Security.

6. View Resources
The chief and agents can view the resources available.

7. Report Management
This feature is provided to the Chief of Security, Agent and the Defense Ministry. The Chief will use this feature to generate reports and send them to the Defense Ministry. The agents can use this feature to send the reports to the Chief. The Defense Ministry will be able to receive the reports.

8. Send Resources to Secret Agency
This feature is provided to the Defense Ministry. The Defense Ministry is responsible for any resources that are to be made available to the agents.

9. Assign Case to Agency
This feature is provided to the Defense Ministry. The defense ministry will create a new case and the case details along with the mission objectives to be sent to agency.

10. View Success Stories
This feature is provided to the Citizen. The citizen has the least powers. The citizen can view the details of completed missions which are posted by the agency.

11. Provide Tips and Feedback
This feature is provided to the Citizen. The citizen can provide tips and feedback regarding any article that is posted by the agency.

12. Apply for Job
This feature is provided to the Citizen. The citizen can inquire about the different job profiles available at with the agency. Also he can inquire about the various qualifications required for different job profiles.

III. BACKGROUND
The Cybernetics Protector software is concerned with the security of the country and thus proper care has to be taken that confidential data from within the database is not leaked out. The main focus of the system is on security and thus the following sets of features are used to provide high security:
- Encryption and Decryption
- RSA Algorithm
- Login
- Case details

A. Encryption and Decryption
This system is based on the 3 pillars of information security- Confidentiality, Integrity and Availability. The digital signature used here protects the integrity and authenticity of a message. However other techniques are still required to provide confidentiality of the message being sent. Encryption is the process of transforming information (referred to as plaintext) using an algorithm (called a cipher) to make it unreadable to anyone except those possessing special knowledge, usually referred to as a key. The result of the process is encrypted information (in cryptography, referred to as cipher text).

To provide higher integrity and confidentiality project uses both the digital signature and encryption mechanisms. The document is digitally signed by the sender as well as the document is encrypted.

B. RSA:
The RSA algorithm was publicly described in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman
The RSA algorithm involves three steps:
- Key generation, encryption and decryption.

Key generation
RSA involves a public key and a private key.
The public key can be known to everyone and is used for encrypting messages.

Messages encrypted with the public key can only be decrypted using the private key.

The keys for the RSA algorithm are generated the following way:
1. Choose two distinct prime numbers $p$ and $q$.
   - For security purposes, the integers $p$ and $q$ should be chosen at random, and should be of similar bit-length.
2. Compute $n = pq$.
   - $n$ is used as the modulus for both the public and private keys. Its length, usually expressed in bits, is the key length.
3. Compute $\phi(n) = (p - 1)(q - 1)$, where $\phi$ is Euler's totient function.
4. Choose an integer e such that 1 < e < φ(n) and greatest common divisor gcd(e, φ(n)) = 1; i.e., e and φ(n) are coprime.
   - e is released as the public key exponent.
   - e having a short bit-length and small Hamming weight results in more efficient encryption – most commonly $2^{16} + 1 = 65,537$. However, much smaller values of e (such as 3) have been shown to be less secure in some settings.[4]

5. Determine d as $d \equiv e^{-1} \pmod{φ(n)}$, i.e., d is the multiplicative inverse of e (modulo φ(n)).
   - This is more clearly stated as solve for d given $de \equiv 1 \pmod{φ(n)}$
   - This is often computed using the extended Euclidean algorithm.
   - d is kept as the private key exponent.

By construction, $d \cdot e \equiv 1 \pmod{φ(n)}$.

The **public key** consists of the modulus $n$ and the public (or encryption) exponent $e$.

The **private key** consists of the modulus $n$ and the private (or decryption) exponent $d$, which must be kept secret. $p$, $q$, and $φ(n)$ must also be kept secret because they can be used to calculate $d$.

- An alternative, used by PKCS#1, is to choose $d$ matching $de \equiv 1 \pmod{λ}$ with $λ = \text{lcm}(p-1, q-1)$, where lcm is the least common multiple. Using $λ$ instead of $φ(n)$ allows more choices for $d$. $λ$ can also be defined using the Carmichael function, $λ(n)$.
- The ANSI X9.31 standard prescribes, IEEE 1363 describes, and PKCS#1 allows, that strong primes, and being different enough that Fermat factorization fails.

### Encryption

Alice transmits her public key $(n, e)$ to Bob and keeps the private key secret. Bob then wishes to send message $M$ to Alice.

He first turns $M$ into an integer $m$, such that $0 ≤ m < n$ by using an agreed-upon reversible protocol known as a padding scheme. He then computes the ciphertext corresponding to $c \equiv m^e \pmod{n}$.

This can be done quickly using the method of exponentiation by squaring. Bob then transmits $c$ to Alice.

### Decryption

Alice can recover $m$ from $c$ by using her private key exponent $d$ via computing $m \equiv c^d \pmod{n}$.

Given $m$, she can recover the original message $M$ by reversing the padding scheme.

(In practice, there are more efficient methods of calculating $c^d$ using the precomputed values below.)

### Using the Chinese remainder algorithm

For efficiency many popular crypto libraries (like OpenSSL, Java and .NET) use the following optimization for decryption and signing based on the Chinese remainder theorem. The following values are precomputed and stored as part of the private key:

- $p$ and $q$ the primes from the key generation,
- $d_P = d \pmod{p - 1}$,
- $d_Q = d \pmod{q - 1}$ and
- $q_{inv} = q^{-1} \pmod{p}$.

These values allow the recipient to compute the exponentiation $m = c^d \pmod{pq}$ more efficiently as follows:

- $\overline{m}_1 = c^{d_P} \pmod{p}$
- $\overline{m}_2 = c^{d_Q} \pmod{q}$
- $h = q_{inv} \cdot (\overline{m}_1 - \overline{m}_2) \pmod{p}$ if $\overline{m}_1 < \overline{m}_2$ then some libraries compute $h$ as $q_{inv} \cdot (\overline{m}_1 + p - \overline{m}_2) \pmod{p}$
- $m = \overline{m}_2 + h \cdot q$

This is more efficient than computing $m = c^d \pmod{pq}$ even though two modular exponentiations have to be computed. The reason is that these two modular exponentiations both use a smaller exponent and a smaller modulus.

### A working example

Here is an example of RSA encryption and decryption. The parameters used here are artificially small, but one can also use OpenSSL to generate and examine a real keypair.

1. Choose two distinct prime numbers, such as $p = 61$ and $q = 53$.
2. Compute $n = pq$ giving $n = 61 \times 53 = 3233$.
3. Compute the totient of the product as $φ(n) = (p - 1)(q - 1)$ giving $φ(3233) = (61 - 1)(53 - 1) = 3120$.
4. Choose any number $1 < e < 3120$ that is coprime to 3120. Choosing a prime number for $e$ leaves us only to check that $e$ is not a divisor of 3120.
   - Let $e = 17$.
5. Compute $d$, the modular multiplicative inverse of $e \pmod{φ(n)}$ yielding $d = 2753$.

The **public key** is $(n = 3233, e = 17)$. For a padded plaintext message $m$, the encryption function is $m^{17} \pmod{3233}$.

The **private key** is $(n = 3233, d = 2753)$. For an encrypted ciphertext, the decryption function is $c^{2753} \pmod{3233}$.
For instance, in order to encrypt $m = 65$, we calculate $c = 65^{17} \equiv 2790 \pmod{3233}$.

To decrypt $c = 2790$, we calculate $m = 2790^{273} \equiv 65 \pmod{3233}$.

Both of these calculations can be computed efficiently using the square-and-multiply algorithm for modular exponentiation. In real life situations the primes selected would be much larger; in our example it would be relatively trivial to factor $n$, 3233, obtained from the freely available public key back to the primes $p$ and $q$. Given $e$, also from the public key, we could then compute $d$ and so acquire the private key.

Practical implementations use the Chinese remainder theorem to speed up the calculation using modulus of factors $(\text{mod } pq \text{ using } \text{mod } p \text{ and } \text{mod } q)$.

The values $d_p$, $d_q$ and $q_{\text{inv}}$, which are part of the private key are computed as follows:

- $d_p = d \pmod{(p-1)} = 2733 \pmod{(61-1)} = 53$
- $d_q = d \pmod{(q-1)} = 2733 \pmod{(33-1)} = 49$
- $q_{\text{inv}} = q^{-1} \pmod{p} = 53^{-1} \pmod{61} = 38$

(Hence: $q_{\text{inv}} \times q \pmod{p} = 38 \times 53 \pmod{61} = 1$)

Here is how $d_p$, $d_q$ and $q_{\text{inv}}$ are used for efficient decryption. (Encryption is efficient by choice of public exponent $e$)

- $m_1 = e^s \pmod{p} = 2790^{59} \pmod{61} = 4$
- $m_2 = e^s \pmod{q} = 2790^{49} \pmod{53} = 12$
- $h = (q_{\text{inv}} \times (m_1 - m_2)) \pmod{p} = (38 \times -8) \pmod{61} = 1$
- $s = s \pmod{1} + h \times 0 = 1\frac{5}{13} + 1 \times 0 = 0\frac{5}{13}$

(same as above but computed more efficiently)

**LOGIN**

The following snapshots of the project screens explain how recognition is implemented and used by different authorities of the Intelligence system.

**CASE DETAILS**

The following snapshots of the project screens explains the case details of the cybernetic protector implemented and used by different authorities of the Intelligence system.