Politecnico di Bari

Telecommunications Engineering

ANALYSIS OF END-TO-END ENCRYPTION IN TELEGRAM

Discrete Mathematics

Professor:
Prof. Angela Aguglia

Student:
Giuseppe Piacenza

Index

Introduction

Smartphones came onto the market in the late 90s, and knew an exponentially increasing popularity among the consumers worldwide. By 2018, over one third of the world's population owns a smartphone, a total of over 2.1 billion smartphone users in the world and sales value of 478.7 billion USD [1] are estimated.

One of the most useful service provided by this technology is the cloud based end-to-end (E2E) instant messaging (IM). A critical problem consists in the privacy of these types of communications, that's why many IM platforms such as Telegram and WhatsApp features E2EE communication.
Telegram is one of the best known cloud based instant messaging (IM) service characterized by its own message encrypting protocol called MTProto which aim is to guarantee secure end-to-end encrypted (E2EE) communications between users. The platform currently has over 200 million monthly active users and 15 billion daily sent text messages, enjoying widespread popularity and greater trust. The company Telegram LLC, founded in 2013 by Nikolai and Pavel Durov, self-describes its product as the best in terms of security and privacy, but not every cryptanalytic expert agrees with this statement. In this report I will describe the communication's mechanisms in Telegram with particoular reference to message encryption using MTProto, discussing and proving

[1] https://www.statista.com/topics/840/smartphones

its strengths and weaknesses against certain types of cryptanalytic attacks.
End-to-end encryption is a mode of communication in which only the communicating parties can read the messages. The content is encrypted through special algorithms thanks to which only the sender and the receiver of the message have the keys to decipher it preventing potential eavesdropper from accessing sensitive data.

The IM platform named Telegram exploits E2EE for its so-called secret chats. Its customized symmetric encryption protocol named MTProto is provided with an official documentation file which is open source so that it is possible to make a detailed analysis of it. In February 2015 Telegram's secret chats received a full rating of 7 out to 7 for its security by the non-profit organization Electronic Frontier Foundation.

The main goal of my report is to discuss Telegram's encryption and decryption mechanisms with particoular reference to communications' security. I will start, in Chapter 1, providing preliminary theoretical concepts that are useful to better understand how MTProto works. Then, in Chapter 2, I will mainly focus on the description of the Telegram's E2EE communication mechanisms starting from the device registration and key exchange using a Diffie-Hellmann (DH) based scheme, pointing out Telegram's advantages in security and privacy. At the end of my report, I'll show a theoretical cryptanalytic attack able to break MTProto.

Chapter 1

Preliminary Concepts

1.1 Modular Arithmetic

Formally, modular Arithmetic consists in a study of operations which are repeated with periodicity. Modular arithmetic is known informally as "clock arithmetic". In modular arithmetic, numbers "wrap around" upon reaching a given fixed quantity, which is known as the modulus. In order to provide a really common example, we can consider the modulus of hours on a clock which is equal to 12.

Definition 1. *Let $a, b \in Z$. The element a is said to be a divisor of b if there exists a $q \in Z$ such that $b = qa$. In this case b is said to be divisible by a. We denote this by $a|b$.*

Properties 1. *The following properties hold $\forall$ $a, b, c \in Z$ with $a \neq 0$:*

1. $a|a$, $1|a$
2. $a|b \longrightarrow a|$ -b
3. $a|b$ *and* $a|c \longrightarrow a|b + c$
4. $a|b$ *and* $b|c \longrightarrow a|c$

5. $a|b$ *and* $b|a \longrightarrow a = \pm b$

Definition 2. *An integer $p > 1$ only divisible by ± 1 or by $\pm p$ is said to be prime.*

We remark that the number 1 is not considered as a prime and the number 2 is the only even prime. The following definition will be useful for further topics:

Definition 3. *A prime number is said to be a safe prime if it is of the form $2p + 1$, where p is also a prime.*

The first few safe primes are 5, 7, 11, 23, 47, 59, 83, 107, ...

Theorem 1 (Division with remainder Property)**.** *$\forall a, b \in Z$, $b \neq 0$, there exists a unique $(q, r) \in Z \times Z$ such that $a = qb + r$ with $0 \leq r <\mid b \mid$.*

1.2 Congruences

It is possible to define a congruence relation as:

Definition 4. *Let $n \in N$, $n \geq 2$. Two integers a and b are said to be congruent modulus n if $n|a$-b. We denote this by $a \equiv b$ mod n. The relation $a \equiv b$ mod n is a congruence relation and the integer n is called the modulus of the congruence.*

Proposition 1. *Two integers are congruent mod n if and only if they have the same remainder when they are divided by n.*

The following theorem is valid for a congruence relation:

Theorem 2. *The congruence relation modulus n is an equivalence relation.*

Proof. The following properties holds:

1. Reflexivity: $\forall a \in Z$ $n \mid a - a$, then $a \equiv a$ mod n.

2. Symmetry: $\forall a, b \in Z$ if $a \equiv b$ mod n, then $n \mid a - b$ and $n \mid b - a$ implies $a \equiv b$ mod n and $b \equiv a$ mod n.

3. Transitivity: $\forall a, b, c \in Z$ if $a \equiv b$ and $b \equiv c$ mod n, then $n \mid a - b$ and $n \mid b - c$ that is, there exist $q, q' \in Z$ such that $(a - b) = qn$ and $(b-c) = q'n$. Hence $(a-c) = (q+q')n$ and $n \mid a-c$. Thus if $a \equiv b \wedge b \equiv c$ mod n then $a \equiv c$ mod n.

□

From the division with remainder property, the following theorem holds:

Theorem 3. *For any integers a and $n \in Z$ with $n \geq 2$, there exists a unique integer $b \in \{0, 1, \ldots, n-1\}$ such that $a \equiv b$ mod n.*

Definition 5. *The equivalence classes of Z with respect to the congruence modulus n are called residue classes.*

We can point out that $\forall a \in Z$, the residue class modulus n of a is the set:

- $[a] = \{b \in Z \mid a \equiv b \text{ mod n } \} = \{a + kn \mid k \in Z\}$

Furthermore if $a \equiv b$ mod n then $[a] = [b]$. The set of all residue classes is denoted by Z_n.
$Z_n = \{[0], [1], \ldots, [n-1]\}$ has as many elements as the possible remainders of the division by n. Then, given the definition of residue classes, let's consider this theorem:

Theorem 4. *Let $a, a', b, b', n \in Z$, $n \geq 2$ such that $a \equiv a'$ and $b \equiv b'$ mod n. Then $a + b \equiv a' + b'$ mod n and $ab \equiv a'b'$ mod n.*

Definition 6. *We can define two operations over Z_n: the addition and the multiplication, as follows:*

- $+ : Z_n \times Z_n \longrightarrow Z_n$:
  $([a], [b]) \longrightarrow [a] + [b] := [a + b]$

- $\cdot : Z_n \times Z_n \longrightarrow Z_n$:
  $([a], [b]) \longrightarrow [a] \cdot [b] := [ab]$

as a consquence of the Theorem 4 the equivalence classes are independent by their representative, then we can point out that the two operations are well defined.

1.3 Groups

In mathematics, a group is considered to be an algebraic structure consisting of a set of elements equipped with an operation that combines any two elements to form a third element and satisfies the so-called group axioms. For further details about groups it's possible to see the references 1 and 2.

Definition 7. *A group consists in a pair (G,*) representing a set G and a binary operation * on G, such that the following properties holds:*

1. $\forall a, b, c \in G$; $a * (b * c) = (a * b) * c$, *then * is associative.*

2. $\exists e \in G$ *such that* $\forall a \in G$, $e * a = a * e = a$. *e is defined as the identity element of the group G.*

3. $\forall a \in G\ \exists x \in G$ *such that* $x * a = a * x = e$. *x is said to be the inverse of a and is denoted as* a^{-1}.

Definition 8. *A group (G,*) is said to be commutative if the binary operation * satisfies:*

- $\forall a, b \in G$, $a * b = b * a$

Examples of commutative groups are (Z, +), (Q, +), (C, +), (R, +) where:

- Z is the set of integers.
- Q is the set of rational numbers.
- C is the set of complex numbers.
- R is the set of real numbers.

Proposition 2. *If G is a group, then there exists a unique identity and each element has a unique symmetric.*

Proof. Lets prove both the statements of the proposition:

1. Identity element uniqueness: suppose $\exists$ e, e' $\in$ G with e and e' both identity elements. Then e=e*e'=e' and the proposition follows.
2. Inverse element uniqueness: suppose $\exists$ x, x' $\in$ G both inverse elements of the element a $\in$ G. Then x=x*e=x*(a*x')=(x*a)*x'=e*x'=x' and the proposition follows.

□

Definition 9. *Let (G,*) be a group and lets denote it by G. The number of elements of G is called the order of G and it is denoted by o(G). If the number of the elements of G is finite, then G is said to be a finite group otherwise it is called a group of infinite order.*

Definition 10. *Let $a \in (G, *)$, where (G, *) is a group, and let n be any integer, then we can define $a^n = a * a * a * a * a \ldots$ (n times) if $n > 0$, $a^0 = e$, and $a^n = (a^{-1})^{-n}$ if $n < 0$.*

Definition 11. *The non-empty subset of G, H, is said to be a subgroup of (G,*) if it is a group using the same binary operation * as is used in G.*

Theorem 5. *Let (G,*) be a group, and let H be a nonempty subset of the set G. H is said to be a subgroup of G if and only if:*

1. $\forall a, b \in H$, $a * b \in H$.
2. $\forall a \in H$, $a^{-1} \in H$

Proposition 3. *If $m, n \in Z$, $\forall a \in G$, where (G, $\cdot$) is a group, then:*

- $a^m \cdot a^n = a^{m+n}$.
- $(a^m)^n = a^{mn}$.

Definition 12. *Let $a \in G$, (G, $\cdot$) is a group, the smallest positive integer n, if it exists, such that $an = 1$ is called the order of a and it is denoted by o(a).*

Given a group G and an element $a \in$G. It's possible to define:

- H = $\{a^m \mid m \in Z\}$

where Z is the set of all integers. H is a subgroup of G. It is the smallest possible subgroup of G which contains the element a, since any group containing a must also contain all the positive and negative powers of a. This subgroup is said to be the subgroup generated by a and we can denote this subgroup by $< a >$. Using the multiplicative notation:

- $< a > = \{e, a, a^2, a^3, \ldots\} = \{x \mid x = a^i \ for \ some \ i \in Z\}$

Using the additive notation:

- $< a > = \{0, a, 2a, 3a, \ldots\} = \{x \mid x = ia \ for \ some \ i \in Z\}$ where:
  1. $na = a + a + a + a \ldots n \ times$
  2. $-na = -a - a - a \ldots -n \ times$

The following theorem holds:

Theorem 6 (Lagrange's Theorem)**.** *If G is a finite group and H is a subgroup of G, then the order of H divides the order of G.*

If G is a finite group and $a \in$G has order n, then the subgroup generated by a has order n and n|o(G). This means that the order of any element of the group is a divisor of the order of the group itself. Let's consider the following theorem:

Theorem 7. *If G is a finite group and $a \in G$, then $a^{o(G)} = e$.*

Definition 13. *A group G is said to be a cyclic group if there exists an element $g \in G$ such that:*

1. $G = \{g^n : n \in Z\}$, *using the multiplicative notation.*
2. $G = \{ng : n \in Z\}$, *using the additive notation.*

The element g is called generator of the group. We can say that $G =< g >$ where $< g >$ is the subgroup generated by g.

Cyclic groups' role is crucial since they may be considered as the fundamental building blocks of finite commutative groups.

Properties 2. *The following properties hold for cyclic groups:*

1. *Every subgroup of a cyclic group is itself cyclic.*
2. *If a generates a cyclic group G of order n, and $b = a^s$, then the cyclic group generated by b has $\frac{n}{gcd(s,n)}$ elements.*

Proposition 4. *Let $n \in N$, where N is the set of natural numbers, $n \geq 2$. Then the set $U_n = \{[x] \in Z_n \mid gcd(x, n) = 1\}$ is a group with respect to the multiplication mod n.*

A complete set of representatives of all distinct elements in U_n is the set of positive integers not exceeding n and coprime to n. Hence, the order of U_n is $\phi(n)$.

Proposition 5. *Considering the set $Z_n \setminus 0$. We indicate it with $Z_n^* = \{1, 2, 3 \dots n-1\}$. It is possible to prove that Z_n^* is a group under multiplication modulo n if and only if n is a prime number.*

Proof. If we consider a composite n we obtain that U_n=$Z_n \setminus \{a : 1 \leq a \leq n, gcd(a, n) > 1\} \supset Z_n^*$. In this case Z_n^* is not a group under multiplication modulo n since this set contains integers $a \in \{1, 2, 3 \dots n-1\}$ with gcd(a, n) > 1. For instance, if p is the smallest prime divisor of n, then $p \in Z_n^*$ but p doesn't belongs to U_n. This means p is not invertible in Z_n^*. So Z_n^* forms a group under multiplication modulo n if and only if n is prime and the largest subset of Z_n that forms a group under multiplication modulo n is U_n. □

Remark 1. *The group Z_p^* with p prime number is cyclic.*

Definition 14. *Two integers are relatively prime (or coprime) if there is no integer greater than one that divides them both (that is, their greatest common divisor is one).*

Definition 15. *The Euler's Totient Function is the function $\phi : N \longrightarrow N$ so defined:*

- $\phi(0)$ *:= 1*
- $\phi(1)$ *:= 1*
- $\phi(n)$ *:=* $\mid \{x \in N \mid x \leq n,\ gcd(n, x) = 1\} \mid$*, if $n > 1$*

Thus $\phi(n)$ is the number of positive integers not exceeding n which are relatively prime to n.

Theorem 8 (Euler's Theorem)**.** *Let $a \in Z$, $n \geq 2$. If gcd(a,n)=1 then the following holds:*

- $a^{\phi(n)} \equiv 1 \ mod \ n$

Proof. Since $\mid U_n \mid = \phi(n)$ we have that $\forall a \in U_n$, $a^{\phi(n)} = 1$, that is, if gcd(a,n)=1 then, $a^{\phi(n)} \equiv 1$ mod n. □

Considering $Z_p \setminus 0 = Z_p^*$, where p is a prime. Z_p^* will be equal to $\{1, 2, 3, \ldots, p-1\}$. Then, we can write the following theorem:

Theorem 9 (Fermat's Little Theorem)**.** *Let p be a prime and let a be any integer. Then, $a^p \equiv a \ mod \ p$.*

Proof. Lets consider the multiplicative group $(Z_p^*, \cdot)$ where p is a prime. It has order $p-1$ and hence $a^{p-1} = 1 \ \forall \ a \in Z_p^*$. Hence, $a^p \equiv a$ mod p $\forall a \in Z$ and the Fermat's Little Theorem follows. □

1.4 Rings and Fields

We can continue the analysis of algebraic structures for cryptological purposes by introducing the concept of ring. A ring consists of a set with two binary operations that generalize the arithmetic operations of addition and multiplication. We can define:

Definition 16. *Let R be a non-empty set and $+$ and $\cdot$ be two operations, called addition and multiplication respectively. Then the triple $(R,+,\cdot)$ is said to be a ring if:*

1. *$(R, +)$ is a commutative group. Its identity will be denoted by 0.*
2. *The operation $\cdot$ is associative.*

3. *Distributivity holds, so that we can state that given (R,+,·) and any elements x, y and z ∈ R it is possible to write* $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$

If · is commutative on R, then R is called commutative ring. If R has the unit-element with respect to · then some element of R may have a symmetric. If a ∈ R has a symmetric, it is unique it is called the inverse of a and is denoted by a^{-1}.

The following theorem holds:

Theorem 10. *Let R be a ring with additive and multiplicative identities 0 and 1 respectively. Then, for all a, b ∈R and any integer n:*

1. *The additive and multiplicative identities are unique.*
2. *0·a=0 (where '0' is the 0-element of the ring on both sides of the equation).*
3. $(-a) \cdot b = -(a \cdot b) = a \cdot (-b)$.
4. $(-a) \cdot (-b) = a \cdot b$.
5. $(n \cdot a) \cdot b = n \cdot (a \cdot b) = a \cdot (n \cdot b)$.

A special subclass of rings is represented by the important structures known as fields. A field consists of a set on which addition, subtraction, multiplication, and division are defined, and behave as when they are applied to rational and real numbers. We can define a field as follows:

Definition 17. *Let F be a non-empty set and + and · be two operations defined on F. Then the triple (F,+,·) is called a field if:*

1. *(F, +) is a commutative group. Its identity is denoted by 0.*

2. *$(F \setminus 0,\cdot)$=$(F^*,\cdot)$ is a commutative group. The multiplicative unit-element is denoted by 1.*

3. *Distributivity holds.*

Definition 18. *It is possible to define the order of a field F, said o(F), as the total number of elements of F. If o(F) is finite, then F is said to be a finite field, otherwise it is a field of infinite order.*

Definition 19. *Let F be a field and m the least positive integer such that:*

- *$m \cdot (1) = 0$*

where 1 is the multiplicative identity of the field F, m is said to be the characteristic of F. If no such m exists, the characteristic is equal to zero.

The following theorem holds:

Theorem 11. *If the characteristic m of a field F is not zero then m is a prime.*

Proof. Suppose that m is not a prime, then $m = ab$, with $a > 1$ and $b > 1$. Now consider the field elements:

- $t = a \cdot (1) = 1 + 1 + 1 + 1 + \ldots a$ times
- $s = b \cdot (1) = 1 + 1 + 1 + 1 + \ldots b$ times

Since $a, b < m$ we have $t \neq 0$ and $s \neq 0$. On the other hand: $t \cdot s = ab \cdot (1) = m \cdot (1) = 0$, which is a contradiction. Thus, m has to be a prime. □

Theorem 12. *If F is a finite field of characteristic m, then F contains m^n elements for some positive integer n. This means that any finite field contains a prime or prime power number of elements.*

Theorem 13. *If F is a finite field, then the multiplicative group of nonzero elements of the field F^*=$F \setminus \{0\}$ is always a cyclic group.*

1.5 Cryptology

Cryptology is the science which provides tools to study and solve issues regarding the privacy of communications. The increasing use in our life of computer controlled communication networks raises problems regarding privacy and authentication. The fundamental goals of cryptography are to provide the following services:

- Confidentiality, that means keeping data secret from all but those who are authorized to see it.
- Authentication, corroborating the source of data, in fact the receiver of a message needs proof that a message comes from a certain party and not from somebody else.

Cryptology can be divided into two disciplines: Cryptography and Cryptanalysis. Cryptography concerns the design of cryptosystems, cryptanalysis studies the breaking of cryptosystems. A cryptanalyst can be:

- Passive, if the cryptanalyst tries to read the transmitted message.
- Active, if the cryptanalyst tries to actively manipulate the data that are being transmitted.

Encryption protects against passive attacks whereas authentication against active attacks: the message is encrypted by the sender using an algorithm. The recipient can decrypt the encrypted message by using the inverse algorithm. For decryption, the recipient has to use a secret, called *key* that gives him an advantage with respect to cryptanalyst. Furthermore a cryptosystem is called:

- Symmetric (or a secret key cryptosystem) if the sender uses the same key utilized by the recipient for the decryption.

- Asymmetric (or a public key cryptosystem) if the sender does not need to know the secret for the decryption.

1.6 Secret Key Cryptosystems

If the message source A wants to send data m to message sink B, he will encrypt it with a cryptographic transformation E_k into c. A and B both know the particular choice of the key by means of a secure chanel. This channel could be a courier, but it could also be that A and B have beforehand agreed on the choice of k. If sender and receiver want to communicate securely with each other both have to agree on the same cryptographic algorithm to use for encrypting and decrypting data. They also have to agree on a common key, the secret key, to use with their chosen encryption/decryption algorithm.

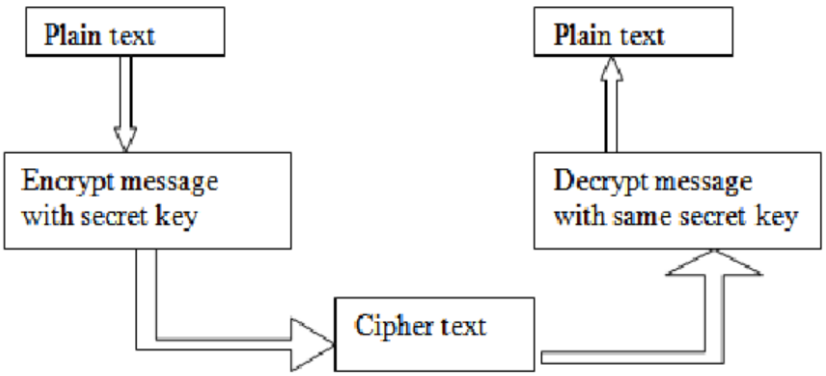

Figure 1.1: Secret Key Cryptosystem scheme.

B can decipher c by computing $E_k^{-1}(c) = E_k^{-1}(E_k(m)) = m$. Secret key encryption and decryption are mathematically inexpensive compared to asymmetric key; therefore, they have a major performance advantage. For any bulk encryption, the preferred method is symmetric encryption.

1.7 Public Key Cryptosystems

Public key cryptosystems are sometimes preferred because of some issue characterizing secret key cryptosystems. In fact, using a conventional cryptosystem, if in a communication system there are n users who all use the same cryptosystem to communicate with each other, it implies a need of $\binom{n}{2}$ keys and $\binom{n}{2}$ secure channels. Whenever a user wants to change his key or a new user wants to participate in the system n - 1 new keys have to be generated and distributed over as many secure channels. Furthermore conventional cryptosystems do not provide the electronic equivalence of a signature. In a public key cryptosystem each user makes his own encryption algorithm E_U and decryption algorithm D_U and for each message m and every user U:

- $D_U(E_U(m)) = m$

Every user U makes his encryption algorithm E_U public by putting it in a key book. The decryption algorithm however, is kept secret by U. If user A wants to send the message m to user B he first looks up the public encryption algorithm E_B of B in the key book. He encrypts m by applying the algorithm E_B to m. Then, he sends c = E(m). User B recovers m from c by applying his secret decryption algorithm D_B to the received cipertext c. We finally obtain:

- $D_B(c) = D_B(E_B(m)) = m$

The algorithms E_U and D_U must satisfy two properties:

1. E_U and D_U must be algorithms with low computational and memory cost.

2. It must be impossible to find an algorithm D_U^* from E_U such that $D_U^*(E_U(m)) = m$ for all possible m.

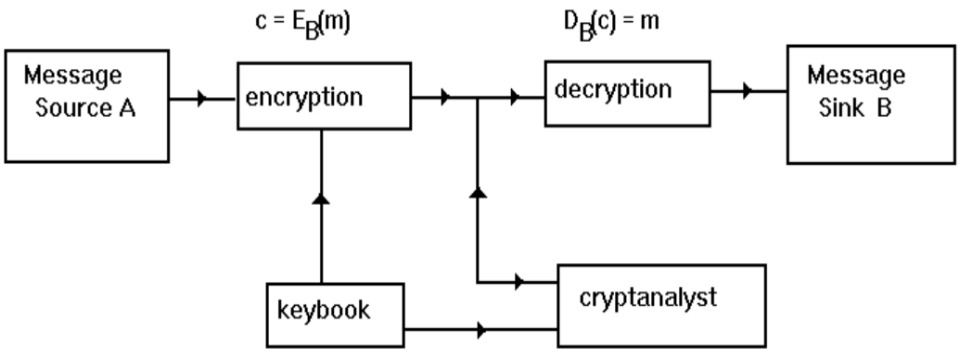

Figure 1.2: Public Key Cryptosystem scheme.

So to cipher, one only has to use the public key of the recipient thus, there is no need of any exchange of keys. The sender does not need to have any secret key, this implies that every user can cipher; just to decipher one needs a secret key that is the secret key of the recipient. Furthermore, if a new user wants to participate in the communication system there is no needs to do any exchange of keys with the older users but the new entry only has to be provided of two keys, the public and the secret one.

1.8 The Discrete Logarithm Problem

Let G be a multiplicative cyclic group generated by an element a, so that the relation G=$< a >$ holds. Let n be the order of G and $h \in$G. The discrete logarithm problem is the problem of finding an integer $x \in \{0, \ldots, n-1\}$ such that:

- a^x=h

That can be written as:

- x=$log_a\ h$

x is called the discrete logarithm of h. There are many different algorithms

able to solve this problem all starting from the same hypothesis but characterized by different efficiencies. In general, the complexity of the Discrete Logarithm Problem (DLP), as we can see in reference 2, depends on the group G. Solving the DLP is crucial to some cryptographic protocol. The solution to the DLP is of great interest to the security of the Diffie-Hellman (DH) protocol.

1.9 The Diffie-Hellman Algorithm

DH is a computationally expensive protocol that has been created to guarantee privacy during communications between users over an insecure channel. The main difficulty of DH protocol consists in computing logarithms over finite fields. Now, let A and B be two users that want to communicate with each other exchanging secret messages. To do this it's necessary that both A and B have a secret key K thanks to which it's possible to decipher every message. The protocol follows five steps:

1. A and B agree over an insecure channel a prime number p, a generator g of Z_p^* so that $g < p$.

2. A generates an integer $1 \leq a < p$ and computes K_A=g^amod p. Then, A sends K_A to B.

3. B generates an integer $1 \leq b < p$ and computes K_B=g^bmod p. Then, B sends K_B to A.

4. The user A computes K_{AB}=K_B^a and B computes K_{BA}=K_A^b.

Now, both A and B share the same secret key K_{AB}=K_{BA}=g^{ab}mod p that has never been sent through the insecure channel. The greatest weakness of DH protocol are Man-in-the-Middle (MITM) cryptanalytic attacks, that will be explained later in this report.

1.10 AES

The Advanced Encryption Standard, or AES, is a symmetric block cipher: it is a method of encrypting text (to produce ciphertext) in which a cryptographic key and algorithm are applied to a block of data at once as a group rather than to one bit at a time. In particoular, the AES is a block cypher with a block size of 128 bits and keysize of either 128 or 192 or 256 bits.

The number of rounds which form the AES depends on the length of blocks and keys and it can be 10, 12 or 14. The plaintext is divided into bytes that are written as columns of an array with 4 rows and a suitable number of columns. There are 4 basic operations when encrypting with the AES.

1. Each byte of the array is mapped to another byte.
2. The elements on each row are cyclically shifted over a given number of positions.
3. Each column is transformed in a new column of byte.
4. Each byte of the last array is added componentwise to a byte of the key given for that round.

The last round doesn't compute the step 3. The most important parameters to evaluate the performance of the AES are security and computational cost.

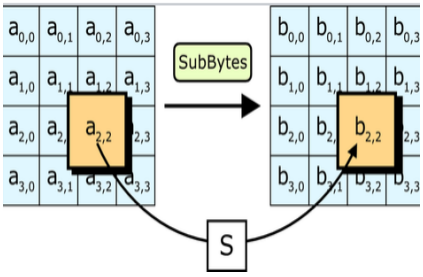

Figure 1.3: Step 1.

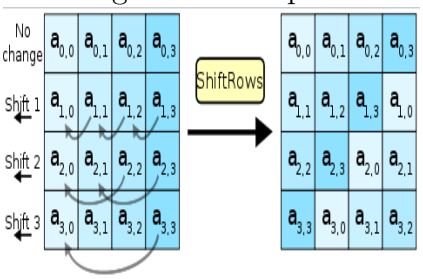

Figure 1.4: Step 2.

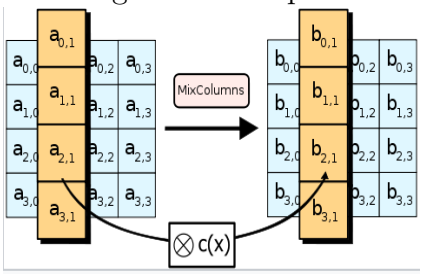

Figure 1.5: Step 3.

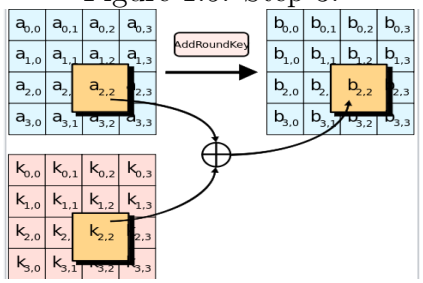

Figure 1.6: Step 4.

1.11 SHA-1

SHA stands for Secure Hash Algorithm. It was designed by the National Security Agency (NSA) to be part of the Digital Signature Algorithm. SHA-1 is a cryptographic hash function that starting from an input message of variable length of maximum 2^{64}-1 bit generates a fingerprint of the message, called *digest* with a fixed length of 160-bit which is designed so that it should be computationaly expensive to find a text which matches a given hash. SHA-1 follows these steps:

1. Create five variables H_0, H_1, H_2, H_3, H_4.

2. Convert the word that we want to hash to ASCII ("American Standard Code for Information Interchange"). Each letter will have a number assigned to it.

3. Convert ASCII code to binary.

4. Merge each binary sequence and add 1 at the end of the construct.

5. Operate a padding with zeros to the sequence to make the message equal to 448 mod 512. This operation is always performed even if the message's length in bit is originally congruent to 448 modulo 512. 448 + 64 = 512, so the message is padded such that its length is now 64 bits less an integer multiple of 512.

6. Add the original message length into the 64 bit field left over after the 448 modular arithmetic.

7. Break the message up into sixteen sections of 32 characters/bits.

8. Through several iterations SHA-1 XoR the words togheter, then it runs a set of functions over the words in a specific order operating off the five

variables that were set in step 1. The functions combine AND, OR and NOT operators combined with left shifts.

9. Convert the H_i variables into hex.

10. Join the variables together to give the hash digest.

Let's see this example, taken from reference 9:

Example 1. *Lets suppose that we want to hash the message "CRYPTO" and we have:*

H0 - 01100111010001010010001100000001

H1 - 11101111110011011010101110001001

H2 - 10011000101110101101110011111110

H3 - 00010000001100100101010001110110

H4 - 11000011110100101110000111110000

The conversion of the chosen word is:

C: 67 $\longrightarrow$ 01000011

R: 82 $\longrightarrow$ 01010010

Y: 89 $\longrightarrow$ 01011001

P: 80 $\longrightarrow$ 01010000

T: 84 $\longrightarrow$ 01010100

O: 79 $\longrightarrow$ 01001111

Then, merging and appending 1 at the end of the sequence:

CRYPTO $\longrightarrow$ 0100001101010010010110010101000001010100010011111

Now, SHA-1 appends 399 zeros on the right and then adds the original message length as a 64 bit block on the right. In this case the length is 48 (1100000). After several operations applied on the sequence the SHA-1 algrithm is able to get the final variables. In this case:

H_0: 01000100101010010111000100110011 $\longrightarrow$ 44a97133

H_1: *01010000111001010011100001011000* $\longrightarrow$ *50e53858*

H_2: *11110000010110000100011000111101* $\longrightarrow$ *f058463d*

H_3: *01001011111101111111000111100101* $\longrightarrow$ *4bf7f1e5*

H_4: *01000010110110011100101001001011* $\longrightarrow$ *42d9ca4b*

Then, the final digest is:

44a9713350e53858f058463d4bf7f1e542d9ca4b

The main advantage of SHA-1 is that minimum changes in the message corresponds to great changes in the final digest. As we can see in the following example [1]:

Example 2. *We consider these words:*

- *Student* $\longrightarrow$ *42b32794792b48313cd1be9ca11b690d3e614683*
- *Students* $\longrightarrow$ *fcfac3efb6ae696fcb48427ff359561932b8f8e8*

[1] http://sha1-hash-online.waraxe.us/

Chapter 2

Telegram connection and security analysis

2.1 Registration to Telegram Server

When a user installs a Telegram's client in a compatible device, this client starts communicating with Telegram's server to create a secret key called authorization key ($auth_{key}$). The authorization key is created by using an authentication protocol that is a modified Diffie-Hellman key distribution algorithm.

Let U be a generic user and S be the Telegram server, the authentication protocol works as follows:

1. U sends a request message req_{pq} to S containing a random string denoted as "nonce".

2. S responds with the res_{pq} message. This message contains another random string said *server nonce*, a composite number n=pq with p and q prime numbers and the fingerprint of the public key for the RSA system.

3. U checks a library of RSA public keys and select the key $server_{PK}$ associated to the received fingerprint. Furthermore, U decomposes n getting

p and q.

4. U creates a new random number, *new nonce*, and sends the message *req_DH_parameters* containing three values encrypted with the Telegram's public key $server_{PK}$.

5. S deciphers the received values and, if the factorization of n is correct, it sends the parameters g, p and g_a=g^amod p. These parameters are encrypted using Advanced Encryption Standard (AES-256) and they are necessary for the application of the Diffie-Hellman protocol. Let's denote this message with *server_DH_params_ok*.

6. U checks that p is a safe 2048-bit prime, that implies $q = \frac{p-1}{2}$ is also a prime, and p is comprised between 2^{2047} and 2^{2048}. Only if every condition is verified, U continues the procedure computing g_b=g^bmod p. Thanks to the knowledge of *new nonce* and *server nonce*, U derives the same AES-256 key generated by S and decrypts g_a.

7. U sends to S the message *set_client_DH_params* containing g_b encrypted with the AES key.

8. S decrypts *set_client_DH_params* and sets $auth_{key}$ to $(g_a)^b$=g^{ab}mod p = g^{ba}mod p, then it sends an acknowledgment signal *dh_gen_ok* to U. The sharing process of the $auth_{key}$ can be considered complete.

The AES-256 encryption of the parameters g, p and g_a makes the protocol robust against Man-in-the-Middle (MITM) attacks, explained in detail in reference 2. In fact, let E be an eavesdropper, in absence of AES encryption of the DH parameters he would be able to pretend to be S with U and U with S. In this case, denoting with E(X) E masquerading as X, the following situation could happen:

1. S sends to E(U) the parameter g^{Xs}.

2. E(S) sends to U the parameter g^{Xe}.

3. U send to E(S) the parameter g^{Xu}.

4. E(U) sends to S the parameter g^{Ye}.

In this way, E could share the secret parameter g^{XsYe} with U who thinks he shares it with S and the secret parameter g^{XeXu} with S who thinks he shares it with U, so that when S and U starts communicating, E could read and re-encrypt every message as shown in Figure 2.1.

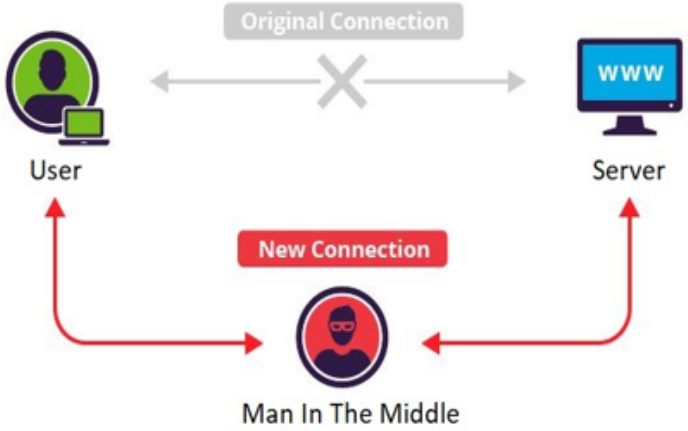

Figure 2.1: Simple MITM attack model.

2.2 Establishing E2EE Communication

Also, E2EE communication between clients is featured by Telegram by exploiting a modified DH protocol. Let A and B be two generic users which want to communicate. The authentication key will be generated as follows:

1. A gets the Diffie-Hellman parameters from the Server: a safe prime p, an integer g and a random sequence x_{ser}. The client checks that both p and $\frac{p-1}{2}$ are primes with p comprised between 2^{2047} and 2^{2048}. If and only if all these conditions are verified, the procedure goes on. It can be useful to store in a cache memory the value of the parameters and the version to avoid having to receive all the values every time a new secret chat is initialized.

2. A generates a random 2048-bit number a, then it computes g_a=g^amod p. The Diffie-Hellman parameter g_a is then sent to client B.

3. B receives the request for the creation of a new secret chat with A. If B accepts, then a new request to the server for DH parameters is done.

4. B computes the same operations of A obtaining g_b=g^bmod p.

5. B generates its $key_{fingerprint}$ that will be used as security check for further key exchanges and sets $auth_{key}$ to $(g_a)^b$=g^{ab}mod p=g^{ba}mod p. Then the couple (g_b, $key_{fingerprint}$) is sent to A.

If either g_a or g_b are not 2048-Bit long a padding procedure is performed, adding several zero bytes until they reach the desired length. The same happens if $auth_{key}$ is not exactly 256 bytes long. The authentication phase in Telegram consists in checking that there are no differences between the encryption keys visualized by A and B. An example of encryption key generated by Telegram is shown in Figure 2.2.

Figure 2.2: Encryption Key visualized on device A.

Telegram uses a re-keying protocol thanks to which clients can initiate re-keying once a key has been used to encrypt and decrypt more than 100 messages or has been in use for more than one week.

2.3 Encryption of an Outgoing Message

The encryption of sent messages is an essential operation for Telegram's secret chats which must be analyzed carefully. As stated above, Telegram uses a proprietary encryption protocol called $MTProto$ which is said to be extremely efficient and secure by the core company themselves. KDF, which stands for Key Derivation Function, is used: it performs several SHA-1 hashes and truncations to obtain the AES key.

The protocol follows a precise procedure:

- A $DecryptedMessage$ packet is created containing the message in plain text and further parameters. It consists of:

Parameter	Basic Bare Type
randomid	long
randombytes	bytes
message	string
media	DecryptedMessageMedia
ttl	int

Where:

1. *randomid* is the identification of the outgoing message.
2. *randombytes* is a set of randomly generated bytes useful to prevent content recognition in short encrypted messages. Messages with less than 15 random bytes are ignored.
3. *message* consists in the text of the outgoing message.
4. *media* describes the file type.
5. *ttl* means *time to live* and it's an arbitrary counter chosen by the sender that specifies the service life of the outgoing message. The ttl is crucial in transmission loops avoidance.

In a Telegram secret chat, there is not an immediate encryption of the messages' plain text; instead, they're previously included in specific packets as shown in Figure 2.3.

- For backward compatibility with previous versions, the packet is included in $decryptedMessageLayer$ which sets the layer number for the contents of an encrypted message. It's made up of:

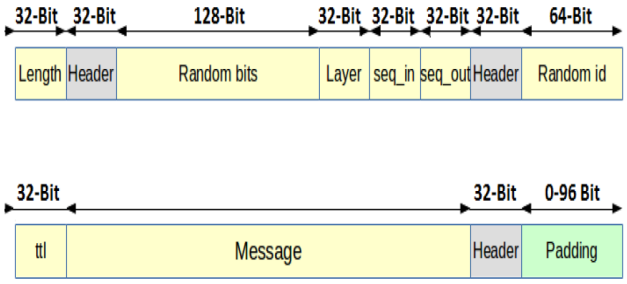

Figure 2.3: Transmitted packet.

Parameter	**Basic Bare Type**
randombytes	bytes
layer	int
in_seq_no	int
out_seq_no	int
message	DecryptedMessageMedia

where:

1. *layer* specifies the layer number starting from a minimal value equal to 17. Layer 17 is the one in which the constructor was added.

2. *in_seq_no* and *out_seq_no* are, respectively, the number of messages received and sent by the chat creator.

- This structure is then serialized as an array of bytes characterized by a total length equal to L. Then, the structure is padded until L is divisible by 16. The header specifies informations regarding the payload's contents such as the protocol version or the type of file while Length specify the total length of the payload.

- The shared 256-bit AES $auth_{key}$ is computed following the steps described in previous paragraphs.

- Encryption key fingerprint ($key_{fingerprint}$) and the message key (msg_{key}) are added at the top of the resulting construct.
- Encrypted data are passed to a *message.sendEncrypted* API call that delivers a text message to a secret chat via Telegram server. The *message.sendEncrypted* API call contains:

Parameter	**Basic Bare Type**
peer	InputEncryptedChat
randomid	long
data	bytes

where:

1. *peer* is the secret chat ID.
2. *randomid* is the unique client message ID, that is necessary to avoid message resending.
3. *data* is the array of bytes encrypted with the AES $auth_{key}$ created during chat initialization.

Similarly, to send an encrypted file in a Telegram's secret chat the file's address must be attached to the outside of an encrypted message. The encrypted file is stored on the server piece by piece using calls to $upload.saveFilePart$, which consists of:

Parameter	**Basic Bare Type**
fileid	long
filepart	int
bytes	bytes

where:

1. $fileid$ is a file identifier created by the client.

2. $filepart$ numbers each part.

3. *bytes* is the content of a part of the file.

Then, a subsequent call to $messages.SendEncryptedFiles$ assigns an identifier to the stored file and sends the address together with the message. The $messages.SendEncryptedFiles$ method works using the following parameters which contains entirely the file informations:

Parameter	**Basic Bare Type**
peer	InputEncryptedChat
randomid	long
data	bytes
file	InputEncryptedFile

This concludes the encryption phase.

2.4 Decryption of an Incoming Message

The decryption process performed by MTProto is basically the reverse of the encryption process, but there are some security checks which deserve to be pointed out.

Before the decryption process starts the receiver checks if the decrypted $auth_{key}$ is equal to $auth_{key'}$ contained in the encrypted message. If $auth_{key}$ $\neq$ $auth_{key'}$ the packet is discarded, otherwise the decryption process can start.

Both $auth_{key}$ and $msg_{key'}$ are passed to the KDF which computes the AES key. Then, if its length doesn't exceed the plain text's one, the msg_{key} is computed using SHA-1. The receiver performs a comparison between msg_{key} and $msg_{key'}$ and only if they're equal the packet is accepted.

2.5 Breaking MTProto

Although spending great efforts on communications' security and reliability, Telegram, just like any other IM service, has bugs and security issues that could be exploited by potential eavesdroppers. In this paragraph, I will discuss a cryptanalytic attack on MTProto which effectiveness can be theoretically proven.

CCA: Chosen Ciphertext Attack

A CCA is a cryptanalytic attack in which the cryptanalyst gathers information by choosing a ciphertext obtaining its corresponding plain text under an unknown key. Then, knowing both the ciphertext and the plain text, the attacker can try to figure out the decryption key.

Ciphertext indistinguishability is the property for which an adversary is unable to distinguish between pairs of ciphertexts based on the message they encrypt. There are three types indistinguishability property:

- Indistinguishability under chosen plaintext attack (IND-CPA)
- Indistinguishability under chosen ciphertext attack (IND-CCA)
- Indistinguishability under adaptive chosen ciphertext attack (IND-CCA2)

A IND-CCA security will also imply IND-CPA security, IND-CCA2 security will also imply both IND-CPA and IND-CCA security.

IND-CCA: Indistinguishability under chosen ciphertext attack

By definition, in IND-CCA the adversary is given access to a *decryption oracle* which decrypts arbitrary ciphertexts at the adversary's request, returning the plaintext. The adversary can query the oracle until it receives the challenge ciphertext.

IND-CCA exploits the fact that in Telegram msg_{key} is computed before padding, without any type of authenticity test on msg_{key} and without checking the padding length. Then, an adversary will be able to add, remove or modify the padding blocks undetected. Let O be the oracle and A be the adversary, IND-CCA follows these steps:

1. A submits two different plaintexts M_0 and M_1 with the same length to O.

2. O selects a bit b$\in$0, 1 and sends the encrypted ciphertext C=Encr(M_b) to A.

3. A adds to C a random 128-bit block c_r and calls O for the decryption of C'=C| | $c_r \neq$ C. It's possible to ask for the decryption of any block that differs from C.

4. O decrypts C' obtaining the payload's length. Knowing the payload's length, A is able to discard the padding bytes. Then, O returns the decrypted M'=Decr(C| | c_r)= M_b.

5. A outputs 1 if M'=M_1 and 0 if M'=M_0.

IND-CCA passes all the security checks performed by MTProto. An adversary A will be able to break MTProto using IND-CCA with probability 1, so it is possible to conclude that MTProto is vulnerable to IND-CCA, in fact a scheme is said to be IND-CCA secure if and only if any A has probability of winning

the decryption challenge: $W<1/2$.

Anyway, this weakness is largely theoretical and still has not been proven practically since it is extremely unlikely that a real adversary is provided with a decryption oracle.

Conclusion

In this report I surveyed the IM platform Telegram describing the modified DH method adopted for key exchange and the encryption/decryption algorithms performed by Telegram's customized communication protocol MTProto.

Then, I discussed MTProto vulnerabilities in terms od IND-CCA. The result of this analysis is that, although being sufficiently secure, MTProto is affected by technical issues that could undermine communication security.

MTProto has never been actually broken, but nevertheless in 2017 Telegram LLC came out with an updated version of MTProto called MTProto 2.0 with many specifications changes in order to enhance the platform's security and reliability, making it, according to Telegram LLC announces, completely secure in terms of IND-CCA.

Finally, my conclusion is that using Telegram is a valuable choice in terms of secure messaging, but users should still be aware of its weaknesses on other types of cryptanalytic attacks.

References

[1] Scott. A. Vanstone, Paul C. van Oorschot, “An introduction to error correcting codes with applications”, Kluwer Academic Publishers, pp. 21–42.

[2] Alko R. Meijer, “Algebra for Cryptologists”, Springer, pp. 45-61, pp. 77-79.

[3] J. D. Vico, “Telegram, bypassing the authentication protocol”, INTECO Cert, pp. 5-7.

[4] Telegram LLC – Official MTProto v1.0 documentation.

[5] T. Susanka, J. Kokes, “Security Analysis of the Telegram IM”, https://www.susanka.eu/files/telegram-article.pdf, pp. 1-9.

[6] J.Jakobsen, C. Orlandi, “A practical cryptanalysis of the Telegram Messaging protocol”, Aarhus University Computer Science Master’s thesis, pp. 37-42.

[7] Indian Institute of Technology Kharagpur – Department of computer science and engineering, FCrypto courses, pp.1-4.

[8] J.Jacobsen, C.Orlandi,“On the CCA (in)security of MTProto”, https://eprint.iacr.org/2015/1177.pdf, pp.4-5.

[9] Antonio Madeira, ”How does a hashing algorithm work?”, Crypto Compare, 20 May 2018.