Criptografia. Uma abordagem inicial com implementação

Victor de Oliveira Guedes[1]

Abstract
Proteção dos dados na Internet é de fundamental importância para as pessoas e um dois meios de se criar tais proteções é através de criptografias. Este artigo visa apresentar os principais fundamentos das criptografias simétricas e assimétricas até os conceitos da criptografia de ponta a ponta. Ao final é descrito a implementação de um software em Android com o uso deste último conceito.

Keywords
criptografia de ponta a ponta — Android — Firebase —

[1] *Mestrado em Sistemas de Informação, Instituto politécnico de Bragança, Bragança, Portugal*
*__Corresponding author__: a40939@alunos.ipb.pt

Contents

Introdução

O termo criptografia tem se tornando comum, uma vez que é empregado cotidianamente principalmente em filmes de ficção científica, entretanto, seu conceito (de como funciona, para que serve e o que realmente é) ainda é abstrato para quem ouve. É necessário saber que a criptografia engloba todas as tecnologias e aplicações e prática e que estas estão inseridas em nossas vidas mais do que imaginamos e seu conceito está ligado intimamente ao conceito de segurança de informação.

Em síntese, vamos entender que a criptografia é um processo de codificação de algo para que não seja facilmente entendido por aqueles que não possuem autorização para acessá-lo. Muitas vezes pode ser enxergado como uma utilização mágica de segurança, que de forma automática faz com que ninguém (exceto o usuário) possa ver o conteúdo, porém em prática não é bem assim. É importante saber que existem vários tipos de criptografia, e que nem todos são difíceis (mas não impossíveis) de descriptografa-los. Estamos acostumados a ver algumas formas simples de criptografia, que consiste em decodificação de troca, que é uma das várias técnicas de criptografias. A mais comum é a troca de Cesar, cujo método consiste em trocar algumas letras do alfabeto para obter o código de descriptografia [1].

Segundo Sautoy [2], o Imperador Romano Júlio César, às vezes usava a técnica de forma que cada letra não criptografada era deslocada em três posições. A Figura 1 feito por Costa [1] exemplifica o processo:

Letra	A	B	C	D	E	F	G	H	I	J	K	L	M
Cifra	C	D	E	F	G	H	I	J	K	L	M	N	O
Letra	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Cifra	P	Q	R	S	T	U	V	W	X	Y	Z	A	B

Figure 1. Troca de César - Fonte de [1]

Nessa configuração, a letra B seria representada por D. Em exemplo, a palavra GUERRA passaria a ser IWGTTC.

Com algumas variáveis, usando qualquer um desses sistemas podemos criar maneiras de cifrar uma mensagem. Independente da escolha, só estamos mudando o símbolo que representa a letra. Dessa forma é fácil compreender como vai se comportar a criptografia só que forma computadorizada.

1. História

Segundo Kahn [3] os primeiros relatos sobre ocultação de mensagens provêm das histórias de Heródoto, filósofo e historiador, que narrou as guerras e conflitos entre a Grécia a Pérsia, durante o século V antes de Cristo. Nesta época o processo mais utilizado era o da esteganografia, que consiste em esconder a mensagem que será enviada [4].

O remetente gravava a mensagem numa tabuleta de madeira e em seguida a cobria com cera, sobre a qual gravava uma nova

mensagem. Ao chegar no destino o remetente, previamente alertado sobre a forma de esconder a mensagem, raspava a cera e lia o conteúdo secreto. No ano de 1563 Blaise de Viginére criou um novo sistema que inicialmente não podia ser criptoanalisado por análise de frequência [3]. O sistema era baseado em mais de um alfabeto e conhecido como método de cifragem poli-alfabético. Em 1854 a Cifra de Viginére é quebrada por Charles Babbage e, por muitos anos, nenhum outro método de criptografia foi desenvolvido de modo a apresentar relativa segurança [5].

Com o desenvolvimento dos computadores mecânicos no início do século XX, máquinas mecânicas de criptografia surgiram e tornaram os processos de criptografia por substituição e transposição mais complexos [6]. Antes do início da Segunda Guerra Mundial, no final da década de vinte, a Alemanha havia desenvolvido uma máquina de criptografia mecânica batizada de Enigma que utilizava métodos já conhecidos de substituição e transposição de forma tão complexa que a criptoanálise manual de suas mensagens era quase impraticável, sendo substituída após a Segunda Guerra Mundial com a evolução da informática, pelos princípios da lógica booleana, desenvolvendo computadores que possibilitaram a criação de algoritmos de substituição e transposição ainda mais complexos de criptoanálise do que a máquina Enigma, sendo necessário criar uma outra máquina, que funcionava com os princípios da lógica de Boole e com as ideias da máquina universal de Turing.

O primeiro computador construído, o Colossus, foi utilizado em 1943 no centro de criptoanálise do Bletchley Park, Inglaterra, para ter a função de analisar códigos, entretanto foi destruído ao final da segunda grande guerra, sendo seus projetos mantidos em segredo e durante muitos anos não foi conhecido [6]. De forma geral os métodos utilizados tinham o mesmo princípio da máquina Enigma. Uma chave era fornecida ao destinatário e em seguida um algoritmo era utilizado para a criptografia. A segurança desse sistema era baseada no fato da chave encontrar-se segura. A esse sistema chamou-se de algoritmos simétricos, pois usavam a mesma chave para criptografar e decriptografar a mensagem transmitida.

Com o tempo houve a necessidade de padronização dos diversos algoritmos de criptografia e, em 1977, a IBM (International Bussiness Machines) criou um algoritmo chamado DES (Data Encryption Standard) que passou a ser utilizado por diversos governos, organizações e empresas como um algoritmo de criptografia padrão. Este algoritmo foi certificado pela NSA (National Security Agency). Com o desenvolvimento tecnológico surgiram computadores capazes de testar milhões de chaves por segundo e que a custos relativamente baixos para grandes empresas e governos, conseguiam quebrar o DES. Em virtude da fragilidade apresentada, o NIST (National Institute of Standards and Technology), através de um concurso, resolveu escolher o substituto do DES, a qual denominariam AES (Advanced Encryption Standard). Uma série de requisitos de segurança foram estabelecidos e o algoritmo escolhido no ano de 2000 foi o Rijndael, segundo Daemem e Rijmen (1999) muitas operações no Rijndael (que foi denominado de AES) são definidas no nível de byte.

2. Criptografia Símétrica e Assimétrica

Os algoritmos simétricos de criptografia foram desenvolvidos e são utilizados até os dias atuais, mesmo possuindo alta vulnerabilidade em sua chave ou senha. Na década de 70 surgiu um novo método criptográfico, o chamado algoritmo assimétrico de criptografia [7], que veio com o intuito de ser aprimorado e garantir maior segurança.

Iniciaremos a explicação por criptografia simétrica, cujo é a mais simples, simplesmente por existir uma única chave entre as operações. Podemos tratar a chave representando um segredo, partilhando entre duas ou mais partes, que podem ser usadas para manter confidencialidade de informação, como podemos perceber na descrição dos algoritmos abaixo:

- DES: Este primeiro exemplo de algoritmo, possui número máximo de 56 bits, é o mais difundido e um dos primeiros também criados. É um tipo de proteção básica que realiza 16 ciclos de codificação para proteger a informação, e para decodificá la precisa usar a mesma chave do início do processo, só que inversamente e pode ser decifrado em horas.
- 3DES: Este segundo, foi o aprimoramento do primeiro, consistindo no mesmo padrão de funcionamento inclusive, porém variando entre 112 a 168 número máximo de bits, cujo executa 3 vezes com 3 chaves diferentes, dificultando para decifrar em até meses.
- AES XEX: Esse algoritmo, por ser mais recente, tem a capacidade de dividir as informações em colunas, para depois realizar uma série de operações misturando estas colunas entre si e utilizando simultaneamente a chave para modificá-la, o que aumenta a segurança, uma vez que esse sistema pode ser decifrado depois de séculos.

Acima, percebemos exemplos de algoritmos de chave simétrica, e notamos também que podem ser tanto divididos como cifras contínuas como de bloco, influenciadas pela quantidade de bits contidas nas mensagens, se vai ser cifrada um a um ou como se fosse uma única unidade. Geralmente os algoritmos de chave simétrica são demorados computacionalmente, na prática significa dizer que os algoritmos assimétricos possuem qualidades superiores à este, ou seja, mais seguro. A desvantagem desse tipo de criptografia simétrica é a exigência de uma única chave secreta compartilhada [8], com cópias apenas na extremidade, e por isso que as chaves necessitam ser trocadas frequentemente, pois dessa forma estão sujeitas à descoberta por um adversário criptográfico.

Diferente do algoritmo simétrico, o algoritmo assimétrico possui a seguinte concepção: são criadas duas chaves, uma pública, que será divulgada para o cifrador, que irá utilizá-la para cifrar a mensagem. Há várias implementações para o esquema de chave pública [7], uma das mais conhecidas

é o RSA, cujo método consiste em um sistema de chaves duplas e na impossibilidade prática de se obter a chave secreta a partir da chave pública. Nos sistemas de chave pública, cada pessoa possui um procedimento para que os outros lhes enviem mensagens criptografadas.

Após a cifragem, somente o decifrador através da segunda chave, a chave privada, poderá decifrar a mensagem. As chaves pública e privada são diferentes e a chave privada não necessita ser transmitida. Esse algoritmo utiliza-se de propriedades das funções unidirecionais da aritmética modular [1].

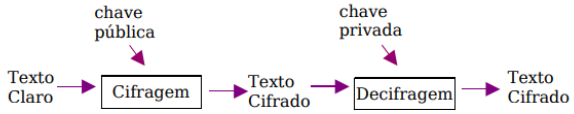

Figure 2. Processo de criptografia. - Fonte de [7]

Os algoritmos de criptografia representam diferentes níveis de proteção, quanto maior a quantidade de bits por chave do algoritmo, mas segura é a codificação. No esquema representado na Figura 2, cada um gera seu par de chaves. Se duas pessoas: A e B, querem se comunicar, cada qual gera seu par de chaves (D_A, E_A) e (D_B,E_B), onde D_A é a chave privada de A, e E_A é sua chave pública, obedecendo a mesma ordem para D_B e E_B. Sendo assim, as chaves públicas podem ser divulgadas e as chaves privadas devem ser mantidas em sigilo. Para mandar a mensagem por exemplo P para B, A criptografa usando E_B, o que resulta no texto criptografando E_B, o que resulta no texto criptografando $E_B(P)$, que é enviado para decifrar a mensagem, B usa sua chave privada DB e recupera o texto inicial $D_B(E_B(P)) = P$DB (EB (P))= P.

Na criptografia moderna, segundo COSTA [1], toda mensagem, ou de maneira geral, qualquer informação é representada por números inteiros. Os processos de cifrar e decifrar são, na verdade, funções que atuam em inteiros. Assim, D_B e E_B são funções matemáticas inversas uma da outra, como mostra a Equação 1:

$$D_B(E_B(P)) = PeE_B(D_B(C)) = C \tag{1}$$

3. Algoritmo Assimétrico

Como já discutido antes, a criptografia assimétrica é conhecida como a criptografia de chave pública e é baseada no uso de pares de chaves. Explanando mais, essas duas chaves são relacionadas através de um processo matemático que usa funções unidirecionais [9] que vai servir para a codificação da informação. Funciona da seguinte forma: uma chave é pública (usada para codificar) e a outra é a chave secreta que servirá para descodificar.

Oliveira [9] explica bem que para entender o conceito desse algoritmo, basta pensar num cadeado comum protegendo um determinado bem. A mensagem é este bem, e o cadeado, que pode ficar exposto, é a chave pública. Apenas quem tiver uma chave particular (privada) que consiga abrir o cadeado poderá acessar a mensagem. A principal vantagem deste método é a sua segurança, pois não é preciso (nem se deve) compartilhar a chave privada. Sendo assim, quando um emissor enviar uma mensagem ao receptor solicitando sua chave pública, o intruso (Man-in-the-middle) poderá interceptá-la e devolver-lhe uma chave pública forjada por ele. Ele também pode fazer o mesmo com o receptor, fazendo com que cada lado pense que está se comunicando com o outro, quando na verdade estão sendo interceptados pelo intruso, então este pode decifrar todas as mensagens [6], cifrá-las novamente ou, se preferir, até substituí-las por outras mensagens. Através deste ataque, um intruso pode causar tantos danos ou até mais do que causaria se conseguisse quebrar o algoritmo de ciframento empregado pelos interlocutores.

Sautoy [2] diz que a garantia para evitar este tipo de ataque é representada pelos certificados de chave pública, comumente chamados de certificado digital, tais certificados consistem em chaves públicas assinadas por uma pessoa de confiança, servindo para evitar tentativas de substituição de uma chave pública por outra. O certificado contém algo mais do que sua chave pública [9], ele apresenta informações sobre o nome, endereço e outros dados pessoais, e é assinado por alguém em quem o proprietário deposita sua confiança, uma autoridade de certificação,sendo definido como um documento eletrônico, assinado digitalmente por uma terceira parte confiável.

Esse sistema também é utilizado como um meio de assinatura digital. A pessoa que assina usa sua chave privada para criptografar uma mensagem conhecida, e o texto cifrado pode ser decifrado por qualquer um usando a chave pública desta pessoa [10], assim como uma assinatura em papel, consiste em um bloco de informação adicionado à mensagem que comprova a identidade do emissor, confirmando quem ele diz ser. O processo se baseia em uma inversão do sistema, onde o funcionamento da assinatura digital pode ser descrito como: o emissor cifra (ou seja, atesta autenticidade) a mensagem com sua chave privada e a envia, em um processo denominado de assinatura digital.

Como a chave pública do emissor apenas verifica a validade das mensagens cifradas com sua chave privada, obtém-se a garantia de autenticidade, o que é apoiado pela função Hashing (algoritmo que mapeia dados de comprimento variável para dados de comprimento fixo) [2], pois se alguém modificar um bit do conteúdo da mensagem ou se outra pessoa assiná-la ao invés do próprio emissor, o sistema de verificação não irá reconhecer a assinatura digital dele como sendo válida. É importante perceber que a assinatura digital, como descrita, não garante a confidencialidade da mensagem. Qualquer um poderá acessá-la e verificá-la, mesmo um intruso, apenas utilizando a chave pública do emissor, assim, ao empregar o uso da técnica de assinatura digital o que se busca é a garantia de autenticidade.

3.1 Rivest, Shamir e Adleman - RSA

Um dos algoritmo mais utilizado e eficiente, isso por que emprega sequências de números primos. O funcionamento por trás do RSA, consiste na fatoração de números primos, ou seja, a facilidade com que o algoritmo pode multiplicar dois números com o intuito de gerar um terceiro sem que seja fácil recuperar os dois primeiros. Em exemplo prático, se possuo um número, por exemplo, 3.337 e eu sei que seus números primos são 47 e 41, a chave pública a ser gerada precisa multiplicar dois números primos grandes, para que a chave privada derive a chave pública. Esse processo qualquer um pode fazer, o segredo está em fatorar um grande número para que não seja de descoberto ou que ao menos leve um tempo maior para que seja decifrada.

Na prática a codificação e decodificação de uma palavra qualquer, por exemplo a palavra "SAIA JÁ", funciona da seguinte forma [1]:

Para codificar, primeiramente é importante definir um par (n;e) = (253;3) que será usado como chave pública. Posteriormente, é necessário achar uma forma de transformar as palavras em números e para isso é possível o uso da Tabela ASCII, com isso a palavra SAIA JÁ torna-se 29111911552011. Com isso é preciso dividir esta sequência em um conjunto de blocos menores que N, onde n é o produto de dois números primos (a explicação detalhada do porquê tal bloco deve ser menor que N será explicado no processo de decodificação). A sequência transformasse em 29 - 111 - 91 - 155 - 201 - 1.

Cada bloco é denotado como C_{b_i}e como $e = 3$ para codificar a mensagem é utilizado a Equação 2. Assim por exemplo o bloco 29 da mensagem deve ser coficado como o resto da divisão de 29^3 por 253.

$$C(b_i) \equiv b_3^i(\mod 253) \tag{2}$$

Assim gerando a mensagem codificada 101-166-137-221-60-1. O que é importante frisar é que estes blocos não podem ser agrupados novamente pois isto pode causar confusão no processo de decodificação.

Já o processo de decodificação é um pouco mais extenso e custoso. Para isso precisa-se dos valores de n, d que é a chave privada, e dois números primos $p = 11$ e $q = 23$ que são os fatores de n. Para encontrar o d utiliza-se da Equação 3

$$d \equiv e^{\phi((p-1)(q-1))-1}(\mod (p-1)(q-1)) \tag{3}$$

O resultado ao final é $d = 27$. Assim o par (n;d) pode ser chamado de chave de decodificação ou chave privada, onde somente quem recebe a mensagem deve obter. Assim, através da Equação 4 é possível achar o valor cifrado. Com esse valor cifrado se n for substituído por p e q é possível chegar ao valor inicial dos blocos [1].

$$D(x_i) = x_i^d(\mod n) \tag{4}$$

Uma chave RSA de 512 bits foi quebrada em 1999 pelo Instituto Nacional de Pesquisa da Holanda, com o apoio de cientistas de mais 6 países. Levou cerca de 7 meses e foram utilizadas 300 estações de trabalho para a quebra. No Brasil, o RSA é utilizado pela ICP-Brasil, no seu sistema de emissão de certificados digitais, e a partir do dia 1º de janeiro de 2012, as chaves utilizadas pelas autoridades certificadoras do país, passam a serem emitidas 5 com o comprimento de 4.096bits, em vez dos 2.048bits atuais

Assim, a segurança do RSA baseia se na dificuldade de fatoração de números grandes. Deste modo, a fatoração representa um limite superior do tempo necessário para quebrar o algoritmo.

3.2 Secure Sockets Layer - SSL / Transport Layer Security - TLS

Desenvolvida pela Netscape, se tornou um padrão de segurança reconhecido pelo cadeado dourado que aparece no browser de internet, que basicamente certifica que um servidor web e navegador estão dentro de um canal criptografado, tornando todo o conteúdo reservado assegurando confiabilidade do website.

Assim, o web site cria as chaves publicas e privadas, a segunda que deve ser mantida em sigilo (como já digo anteriormente) e a primeira pode ser compartilhada na CSR (Certificate Signing Request) que é o arquivo onde está presente os conteúdos do Web site. Este arquivo é enviado ao para os browser para assim criar o certificado digital do cliente.

Esta conexão é estabelecido através de um funcionamento handshake como pode ser visto pela Figura 3, onde o cliente envia a solicitação de que quer uma conexão segura, o servidor responde enviando o arquivo com a chave pública, o cliente autentica seu certificado e gera uma chave pública simétrica randomicamente criptografando-a com a chave pública do servidor. Agora ambos conhecem a chave um do outro, protegendo os dados nas transações [11].

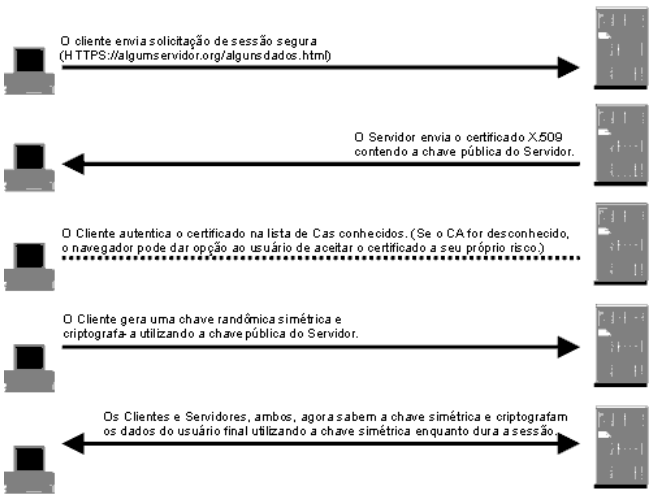

Figure 3. Handshake SSL [11]

Os algoritmos de criptografia simétricos e funções hash

que podem ser usados pelo SSL / TLS são:

- DES e 3DES;
- RC2 e RC4;
- Skipjack;
- IDEA;
- AES;
- MD5 e SHA-1.

Os algoritmos de criptografia assimétricos e funções hash que podem ser usados pelo SSL / TLS são:

- DSA e RSA são usados verificação de assinatura;
- RSA é usado para trocar de chaves

4. Criptografia de ponta a ponta

Frequentemente percebemos ainda mais a inserção da criptografia no nosso cotidiano, encontrando principalmente no que se refere aos meios de comunicação e dispositivos eletrônicos. A partir de tudo que foi citado até o presente momento, entendemos que com um simples pagamento feito com um cartão de crédito, uma ligação por telefone, bilhete eletrônico de ônibus, logon de computador, o acesso ao e-mail ou a um site seguro, a criptografia está presente [10].

A criptografia de ponta a ponta é um dos conceitos presentes neste contexto e esta tem como caracteristica o conceito de que os clientes são responsáveis pela codificação e decodificação dos dados, onde o servidor central serve apenas para encamarinhar esse dados aos seus respectivos clientes. A Figura 4 mostra um exemplo de como esta transição acontece. Também é recomendado a autenticação dos usuários para validação se aquele usuário é realmente quem diz ser, e criar um túnel SSL/TLS.

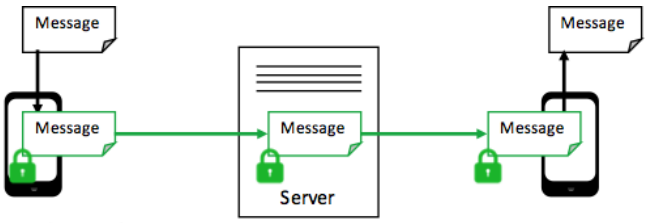

Figure 4. Criptografia de ponta a ponta

O whatsapp, aplicativo de comunicação, que popularizou em 2016 a criptografia de ponta-a-ponta, assegurando seus usuários que somente o usuário e o destinatário da mensagem possam ter acesso ao que foi enviado/recebido e ninguém mais, nem mesmo o whatsapp, o que tornou propaganda principal em todas as plataformas que se acessa. Essa nova atualização do aplicativo acontece de forma automática, não sendo necessário o usuário ativar as configurações ou estabelecer tipos de conversas secretas, isso por que o próprio aplicativo já fornece chaves especiais necessárias para destrancá-la e ler, não tornando possível também o desativo dessa função.

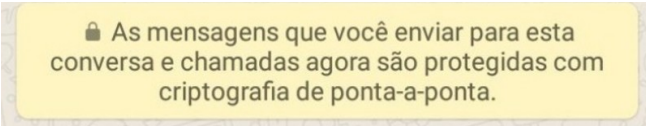

Figure 5. Mensagem informativa em todas as mensagens. - Fonte autor

O gerenciamento de chaves passa a ter dois novos aspectos nesse sistema de chave pública: o primeiro, deve-se previamente localizar a chave pública de qualquer pessoa com quem se deseja comunicar e, segundo, deve-se obter uma garantia de que a chave pública encontrada seja proveniente daquela pessoa. Sem esta garantia, um intruso pode convencer os interlocutores de que chaves públicas falsas pertencem a eles [9].

Estabelecendo esse processo de confiança entre os interlocutores, o intruso pode fazer-se passar por ambos, porém, propaganda da segurança padrão do whatsapp consiste em assegurar o usuário que fale livremente, evitando vazar informações, deixando o usuário a vontade através de mensagens e ligações livres, assegurando também que podemos confirmar a criptografia, como mostra a Figura 6:

Figure 6. Confirmação código de segurança do whatsapp. - Fonte autor

Tecnicamente o processo de trocas de mensagens pelo whatsapp é dado que uma vez que seção é estabelecida entre os usuários, a troca é protegida com o uso do algoritmo de criptografia AES256 e com o HMAC-SHA256 para a autenticação dos usuários. A chave da mensagem é alterada em toda nova mensagem transmitida ou recebida, isso garante que em esta não seja reconstruída [12].

Os usuários do WhatsApp também têm a opção de verificar as chaves com quem eles estão se comunicando para que eles sejam capazes de confirmar que um terceiro não autorizado (ou o próprio WhatsApp) [12], que contém digitalizado um código QR ou comparando em número, uma sequência numérica de 60 dígitos. O código QR contém: 1) Uma versão; 2) O identificador de usuário para ambas as partes; 3) A chave

de identidade pública completa de 32 bytes para ambas as partes.

4.1 Exemplo de implementação

É claro que não é divulgado o código fonte de como o WhatsApp implementa essa criptografia, por isso nessa subseção será demonstrado um exemplo de implementação de um chat com criptografia de ponta a ponta usando o mesmo algoritmo AES256. Este que deve ser levado como exemplo, dependendo do problema pode não ser a melhor solução.

Para realizar essa tarefá foi utilizado o Android com a linguagem Kotlin e Java do lado do cliente, e no lado do servidor a ferramenta Firebase, que é recurso desenvolvido pela Google para ajudar e abstrair o processo de criação de Webservices, fazendo com que o desenvolvedor foque principalmente na aplicação e no seu negócio. O Firebase também utiliza de SSL/TLS para proteção dos dados[13]. O Resultado final da aplicação pode ser visto na Figura 7

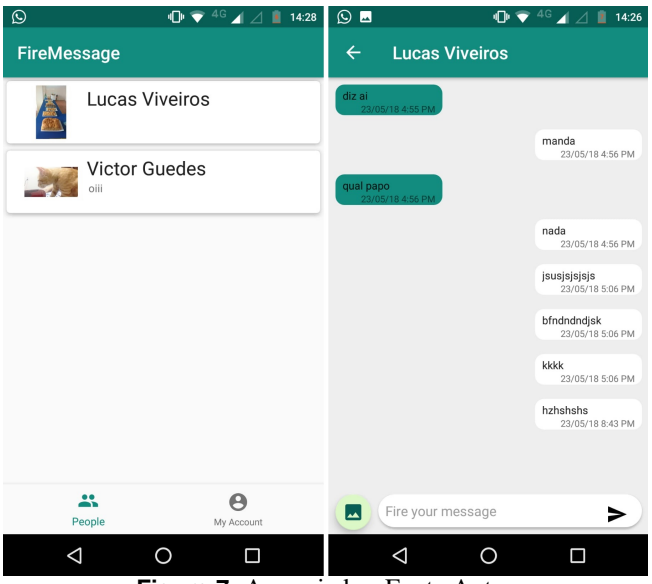

Figure 7. App criado - Fonte Autor

Claramente ”reinventar a roda” não é o recomendado, assim foi utilizado a Classe Cipher e a api AESCrypt-Android [14] do Android para desenvolver a codificação e decodificação das mensagens da aplicação.

Toda vez que é aberto um Chat é criado uma chave que será compartilhado entre as duas pessoas da conversa. O processo de encriptação da mensagem é dado pelo código da Figura 8. Onde para toda mensagem encriptada é gerado uma chave com base na chave do chat. Com isso é encriptado o dado retornando uma String para ser enviado ao servidor. Para o código é utilizado os seguintes atributos:

- AES_MODE = AES/CBC/PKCS7Padding;
- CHARSET = UTF-8;
- HASH_ALGORITHM = SHA-256;
- ivBytes = 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00

- este que não é a melhor solução, mas se encaixa no problema.

```
private static SecretKeySpec generateKey(final String password) throws NoSuchAlgorithmException, UnsupportedEncodingException {
    final MessageDigest digest = MessageDigest.getInstance(HASH_ALGORITHM);
    byte[] bytes = password.getBytes("UTF-8");
    digest.update(bytes, 0, bytes.length);
    byte[] key = digest.digest();
    SecretKeySpec secretKeySpec = new SecretKeySpec(key, "AES");
    return secretKeySpec;
}

public static String encrypt(final String password, String message)
        throws GeneralSecurityException {

    try {
        final SecretKeySpec key = generateKey(password);
        byte[] cipherText = encrypt(key, ivBytes, message.getBytes(CHARSET));
        String encoded = Base64.encodeToString(cipherText, Base64.NO_WRAP);
        return encoded;
    } catch (UnsupportedEncodingException e) {
        if (DEBUG_LOG_ENABLED)
            Log.e(TAG, "UnsupportedEncodingException ", e);
        throw new GeneralSecurityException(e);
    }
}

public static byte[] encrypt(final SecretKeySpec key, final byte[] iv, final byte[] message)
        throws GeneralSecurityException {
    final Cipher cipher = Cipher.getInstance(AES_MODE);
    IvParameterSpec ivSpec = new IvParameterSpec(iv);
    cipher.init(Cipher.ENCRYPT_MODE, key, ivSpec);
    byte[] cipherText = cipher.doFinal(message);
    return cipherText;
}
```

Figure 8. Encriptação [14]

Assim o servidor recebe a mensagem encriptada e encaminha para o cliente respectivo. O servidor também armazena a mensagem, mas está não pode ser lida. O processo de decodificação de mensagem acontece apenas nos clientes, que pode ser lido pelo código da Figura 9. Basicamente é passado a mensagem encriptada e a chave do chat. Assim é decriptada a mensagem e o retorno em texto puro.

```
public static String decrypt(final String password, String base64EncodedCipherText)
        throws GeneralSecurityException {

    try {
        final SecretKeySpec key = generateKey(password);
        byte[] decodedCipherText = Base64.decode(base64EncodedCipherText, Base64.NO_WRAP);
        byte[] decryptedBytes = decrypt(key, ivBytes, decodedCipherText);
        String message = new String(decryptedBytes, CHARSET);

        return message;
    } catch (UnsupportedEncodingException e) {
        if (DEBUG_LOG_ENABLED)
            Log.e(TAG, "UnsupportedEncodingException ", e);

        throw new GeneralSecurityException(e);
    }
}

public static byte[] decrypt(final SecretKeySpec key, final byte[] iv, final byte[] decodedCipherText)
        throws GeneralSecurityException {
    final Cipher cipher = Cipher.getInstance(AES_MODE);
    IvParameterSpec ivSpec = new IvParameterSpec(iv);
    cipher.init(Cipher.DECRYPT_MODE, key, ivSpec);
    byte[] decryptedBytes = cipher.doFinal(decodedCipherText);

    return decryptedBytes;
}
```

Figure 9. Decodificação [14]

Para mais informações de como foi implementado o app e suas particularidades de código, este pode ser acessado no endereço `https://github.com/VictorGuedes/App`.

5. Conclusão

No mundo atual onde quase todas as informações pessoais são praticamente abertas, manter proteção de tais dados é quase que uma obrigação por parte das empresas. Portanto, mesmo que difícil de se implementar já existem soluções prontas para que o desenvolvedor faça do seu sistema ou negócio o mais seguro possível para os seus usuários. Mesmo assim ainda ocorrerá casos de roubo de dados, invasões de sistemas, falhas, o que é bom, pois essas tecnologias se manterão em constante crescimento.

References

[1] Darci Costa. A matemática e os códigos secretos: uma introdução à criptografia. *PROFMAT*, 2014.

[2] Marcus Du Sautoy. A música dos números primos: a história de um problema não resolvido da matemática. *Tradução, Diego Alfaro. Rio de Janeiro: Zahar*, 2007.

[3] The CodeBreakers. page 473.

[4] Bernardino SANT' ANA JÚNIOR. Introdução à matemática aplicada à criptologia. *Trabalho de Conclusão de Curso (Licenciatura em Matemática) - Universidade Castelo Branco, Rio de Janeiro*, 2004.

[5] Simon SINGH. O livro dos códigos. *Rio de Janeiro: Record*, 1999.

[6] Marcelo Ferreira ZOCHIO. Introdução à criptografia. *Novatec Editora LTDA*, 2016.

[7] Figueredo. *Introdução à criptografia.*, volume 2. 2010.

[8] Behrouz A. FOROUZAN. *Comunicação de dados e Redes de computadores.* 2010.

[9] Ronielton Rezende OLIVEIRA. Criptografia tradicional simétrica de chave privada e criptografia assimétrica de chave pública: análise das vantagens e desvantagens. *Trabalho da pós-graduação Criptografia e Segurança em Redes da UFF, Niteroi*, 2006.

[10] Sean Michael WYKES. Criptografia essencial: A jornada do criptógrafo. *Elsevier Editora LTDA*, 2016.

[11] Como funciona o ssl. `http://publib.boulder.ibm.com/tividd/td/TRM/GC32-1323-00/pt_BR/HTML/admin231.htm`. Accessed: 2018-05-24.

[12] WhatsApp. Whatsapp encryption overview technical white paper. 2017.

[13] Server-side encryption. `https://cloud.google.com/firestore/docs/server-side-encryption`. Accessed: 2018-05-20.

[14] Aescrypt-android. `https://github.com/scottyab/AESCrypt-Android`. Accessed: 2018-05-20.