TIAGO ROSSI DIAS

CRIPTOGRAFIA COM RESÍDUOS QUADRÁTICOS

Santo André, 2017

UNIVERSIDADE FEDERAL DO ABC

CENTRO DE MATEMÁTICA, COMPUTAÇÃO E COGNIÇÃO

TIAGO ROSSI DIAS

CRIPTOGRAFIA COM RESÍDUOS QUADRÁTICOS

Orientador: Prof. Dr. Jerônimo Cordoni Pellegrini

Dissertação de mestrado apresentada ao Centro de Matemática, Computação e Cognição para obtenção do título de Mestre

ESTE EXEMPLAR CORRESPONDE A VERSÃO FINAL DA DISSERTAÇÃO
DEFENDIDA PELO ALUNO TIAGO ROSSI DIAS,
E ORIENTADA PELO PROF. DR. JERÔNIMO CORDONI PELLEGRINI.

SANTO ANDRÉ, 2017

Sistema de Bibliotecas da Universidade Federal do ABC
Elaborada pelo Sistema de Geração de Ficha Catalográfica da UFABC
com os dados fornecidos pelo(a) autor(a).

Dias, Tiago Rossi
Criptografia com Resíduos Quadráticos / Tiago Rossi Dias. — 2017.

34 fls.

Orientador: Jerônimo Cordoni Pellegrini

Dissertação (Mestrado) — Universidade Federal do ABC, Mestrado Profissional em Matemática em Rede Nacional - PROFMAT, Santo André, 2017.

1. Criptografia. 2. Teoria dos Números. 3. Resíduos Quadráticos. I. Pellegrini, Jerônimo Cordoni. II. Mestrado Profissional em Matemática em Rede Nacional - PROFMAT, 2017. III. Título.

Este exemplar foi revisado e alterado em relação à versão original, de acordo com as observações levantadas pela banca no dia da defesa, sob responsabilidade única do autor e com a anuência de seu orientador.

Santo André, 12 **de** maio **de 20**17**.**

Assinatura do autor: _____

Assinatura do orientador: _____

MINISTÉRIO DA EDUCAÇÃO
Fundação Universidade Federal do ABC
Programa de Pós-Graduação em Mestrado Profissional em Matemática em Rede Nacional
Avenida dos Estados, 5001 – Bairro Santa Terezinha – Santo André – SP
CEP 09210-580 · Fone: (11) 4996-0017
profmat@ufabc.edu.br

FOLHA DE ASSINATURAS

Assinaturas dos membros da Banca Examinadora que avaliou e aprovou a Defesa de Dissertação de Mestrado do candidato Tiago Rossi Dias, realizada em 17 de fevereiro de 2017:

Prof.(a) Dr.(a) **Jerônimo Cordoni Pellegrini** (Universidade Federal do ABC) – Presidente

Prof.(a) Dr.(a) **Sinue Dayan Barbero Lodovici** (Universidade Federal do ABC) – Membro Titular

Prof.(a) Dr.(a) **Alexandre Lymberopoulos** (Universidade de São Paulo) – Membro Titular

Prof.(a) Dr.(a) **Eduardo Guéron** (Universidade Federal do ABC) – Membro Suplente

Prof.(a) Dr.(a) **Paola Andrea Gaviria Kassama** (Universidade Federal de São Paulo) – Membro Suplente

Dedico este trabalho a meu pai, que mesmo longe ainda é meu herói, e a minha mãe, que amo incondicionalmente.

AGRADECIMENTOS

Agradeço a Deus, por tudo que Ele possibilitou em minha vida, principalmente por ter tido a dádiva de conhecer a todos que agradecerei a seguir.

Agradeço a meu pai e minha mãe, por mostrarem para mim o quão importante é o estudo e especialmente incentivar o gosto que tenho pela Matemática.

Agradeço a minha irmã e todos os meus parentes e amigos, pelo apoio e suporte que me deram.

Agradeço a equipe gestora da Escola Estadual Prefeito Domingos de Souza, pela ajuda que deram para que eu pudesse cursar esse mestrado.

Por último, mas não menos importante, agradeço a todos os meus professores, pois sempre me incentivaram a questionar e buscar o conhecimento.

"A Matemática, devidamente observada, possui não somente a verdade, mas suprema beleza - uma beleza fria e austera, como a de uma escultura."

(Bertrand Russel)

RESUMO

Esse trabalho tem como objetivo mostrar como problemas de difícil solução, em especial o problema dos resíduos quadráticos, podem ser usados para desenvolver criptossistema com segurança demonstrável, com algumas aplicações que podem ser desenvolvidas com alunos de ensino fundamental e médio. Faz-se um resumo da história da criptografia, desde a Cifra de César e passando por diversos criptossistemas historicamente famosos, até chegar ao sigilo perfeito do one-time pad. São trabalhados também alguns conceitos matemáticos necessários, como as funções de mão única e uma breve explicação de algumas funções conjecturadas de mão única, que podem ser usadas em sistemas criptográficos seguros. Em seguida, apresenta-se os geradores de números pseudo-aleatórios, em especial o de Blum-Blum-Shub por empregar resíduos quadráticos. A seguir, há uma breve apresentação das funções de hash e do problema do aniversário associado a elas, com uma função de hash construída baseada no gerador de Blum-Blum-Shub. Também importante é a aplicação na encriptação com chave pública, em especial o criptossistema de Rabin, que também é usado para estabelecer um sistema de votação com base no homomorfismo apresentado por esse sistema. Para finalizar, fala-se sobre as provas de conhecimento zero e como as raízes quadradas módulo N podem ser utilizadas para isso, em particular com o Protocolo de Feige-Fiat-Shamir. Uma aplicação para a sala de aula é dada na forma de um leilão, utilizando o conceito da dificuldade da raiz quadrada modular.

Palavras-chave: criptografia, teoria dos números, resíduos quadráticos

ABSTRACT

The main objective of this work is to show how hard to solve problems, specially the problem of quadratic residuality, can be used to create cryptographic algorithms with provable security. Some applications could be done with students from elementary and high school. We will start with a brief history of cryptography, from Cesar Cipher and going through several famous cryptosystems until the perfect secrecy of the one-time pad. We will work in a few basic concepts, such as one-way functions and a succinct explanation on some functions that are conjectured to be one-way and can be used in provably secure cryptographic systems. We choose the modular squaring to show on the following chapters how one-way functions are used to build several algorithms (pseudo-random number generators, hash functions, public key encryption, a voting system based on a homomorphic cryptosystem and, at last, zero-knowledge proofs). We will provide a classroom example in the ways of an auction, using the difficulty of the modular square root.

Keywords: cryptography, number theory, quadratic residue

CONTEÚDO

INTRODUÇÃO 1

1 HISTÓRIA DA CRIPTOGRAFIA 3
- 1.1 A Cifra de César . . . 3
- 1.2 O Atbash Hebraico . . . 4
- 1.3 A Cítala Espartana . . . 4
- 1.4 A Cifra de Vigenère . . . 5
- 1.5 A Cifra de Hill . . . 7
- 1.6 A Máquina Enigma . . . 8
- 1.7 O Princípio de Kerckhoff . . . 8
- 1.8 Sigilo Perfeito . . . 9
- 1.9 One-time pad . . . 10
- 1.10 Segurança Demonstrável . . . 11

2 FUNÇÕES DE MÃO ÚNICA 13
- 2.1 Função Desprezível . . . 13
- 2.2 Função de Mão Única . . . 14
  - 2.2.1 Exponenciação Discreta . . . 15
  - 2.2.2 Multiplicação de Inteiros . . . 15
  - 2.2.3 Quadrado Modular . . . 16
  - 2.2.4 Soma de Subconjuntos . . . 17
- 2.3 Predicado hard-core . . . 17

3 GERADORES DE NÚMEROS PSEUDO-ALEATÓRIOS 19
- 3.1 Gerador de Números Pseudo-aleatórios de Blum-Blum-Shub . . . 20

4 FUNÇÕES DE HASH 23
- 4.1 Requisitos das Funções de Hash . . . 23
- 4.2 Problema do Aniversário . . . 25
- 4.3 Construção de Função de Hash . . . 26
- 4.4 Resíduos Quadráticos como Função de Hash . . . 26

5 ENCRIPTAÇÃO COM CHAVE PÚBLICA 29
5.1 Criptossistema de Rabin 30
5.2 Segurança do Criptossistema de Rabin 32
5.3 Encriptação Homomórfica 33
5.3.1 Sistemas de Votação 33

6 PROVA DE CONHECIMENTO ZERO 37
6.1 Prova de Conhecimento Zero com Raízes Quadradas Módulo N 38
6.2 Protocolo Feige-Fiat-Shamir 39

7 APLICAÇÃO EM SALA DE AULA 43

Bibliografia 47

INTRODUÇÃO

A criptografia é um campo importante do conhecimento humano, principalmente na era atual, onde tantas informações importantes trafegam diariamente pela internet. Entre outros méritos, a criptografia trouxe uma aplicação para a área da Teoria dos Números que era considerada uma das áreas mais abstratas da Matemática. Mesmo com a importância atual dessa parte da Matemática, ela é pouco trabalhada nas escolas, que enfatizam o conjunto dos números reais para tratamento e solução de problemas.

Como veremos, questões importantes e de difícil solução podem ser encontradas no campo da Matemática Discreta, podendo citar os resíduos quadráticos dentro das classes de congruências módulo m. De maneira geral, dizemos que a é um resíduo quadrático módulo p se a congruência $X^2 = a \pmod{p}$ tem solução; caso contrário, dizemos que a não é resíduo quadrático módulo p.

Com esse trabalho, procuramos encontrar um sistema criptográfico simples o suficiente para poder ser trabalhado com alunos de ensino médio mas com segurança demonstrável a partir de conjecturas confiáveis, entre elas a dificuldade da fatoração de inteiros e a raiz quadrada modular.

No decorrer deste trabalho, mostraremos como os resíduos quadráticos podem ser usados em alguns criptossistemas, com exemplos práticos que podem ser adaptados para outras situações. Certamente, existem outros meios de obter os resultados desejados com métodos computacionalmente mais rápidos, mas o conceito matemático serve como base para aprofundar os conhecimentos sobre o tema proposto.

Esse trabalho tem como objetivo mostrar aplicações para a matemática modular, em especial os resíduos quadráticos, por tratar-se de um campo meio esquecido nos ensinos fundamental e médio que costumam priorizar a matemática com números reais mesmo em situações onde faz mais sentido trabalhar com números inteiros.

Nesse estudo, veremos um método interessante para votação usando o problema da raiz quadrada modular, utilizando para sua segurança um problema conjecturado de difícil solução. Além disso, mostramos um estudo sobre outros sistemas criptográficos importantes que utilizam resíduos quadráticos para demonstrar sua segurança.

Para desenvolver esse trabalho, foi feita uma pesquisa bibliográfica com diversas fontes, buscando textos e artigos importantes da criptografia, alguns ainda usados atualmente para segurança de sistemas.

O trabalho está dividido em sete capítulos, começando com uma revisão histórica da criptografia e depois fazendo uma breve descrição sobre funções de mão única. Em seguida, falamos sobre geradores pseudo-aleatórios, em especial o de Blum-Blum-Shub e depois sobre funções de hash. Há um capítulo sobre sistemas de chave pública, onde não poderia faltar o criptossistema de Rabin e em seguida uma aplicação interessante dos resíduos quadráticos com provas de conhecimento zero. Finalizamos com uma aplicação em sala de aula interessante, que pode ser usada com alunos de ensino fundamental e médio após alguma preparação.

1

HISTÓRIA DA CRIPTOGRAFIA

É difícil dizer com certeza quando surgiu a criptografia. Sabe-se que, com o advento da escrita, surgiu a necessidade de impedir que determinados textos fossem lidos. Esses textos provavelmente deveriam conter segredos de estado ou planos de batalha.

Podemos dividir a história da criptografia em três fases [6]. Na primeira fase, mais antiga, os textos a serem criptografados usavam apenas papel e tinta para serem cifrados e decifrados. A segunda fase chega com a utilização de máquinas eletro-mecânicas na criptografia, na época da Segunda Guerra Mundial. A terceira fase é a atual, com os computadores fazendo o trabalho criptográfico. Nesse último período, mais atual, a Matemática passa a ter um papel importante devido ao conceito de segurança demonstrável.

Nesse trabalho serão apresentados apenas alguns exemplos de cifras, pois uma descrição completa está fora do escopo desse trabalho. Para os interessados em uma descrição mais detalhada, ver [6] ou [12].

1.1 A CIFRA DE CÉSAR

Essa cifra, creditado ao imperador romano Júlio César, é relativamente simples de cifrar e decifrar. Nele as letras do texto original (em minúsculas) eram trocadas pelo texto criptografado (em maiúsculas) de acordo com a tabela:

a	b	c	d	e	f	g	h	i	j	k	l	m
↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
D	E	F	G	H	I	J	K	L	M	N	O	P

n	o	p	q	r	s	t	u	v	w	x	y	z
↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Por exemplo, o texto “hoje e segunda feira” seria transformado em “KRMH H VHJXQGD IHLUD”. Os espaços estão aí apenas para melhorar a compreensão, em um texto criptografado eles seriam retirados.

Esse tipo de cifra é chamado de cifra de deslocamento. Se cada letra de A a Z for transformada em um número inteiro correspondente de 0 a 25, essa transformação é análoga a operação de adição modular, onde o texto original d é transformado no texto cifrado c usando a fórmula $c = d + k \pmod{26}$. No exemplo da cifra de César, temos que a chave k vale 3, mas poderia ser qualquer outro número inteiro. Isso fornece 25 chaves diferentes (excluindo-se o 0, que tornaria o texto cifrado igual ao original).

1.2 O ATBASH HEBRAICO

No atbash, a substituição é feita substituindo a primeira letra do alfabeto pela última, a segunda pela penúltima e assim sucessivamente. Em nosso alfabeto, teríamos a seguinte tabela de substituição:

a	b	c	d	e	f	g	h	i	j	k	l	m
↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕	↕
z	y	x	w	v	u	t	s	r	q	p	o	n

Duas coisas chamam a atenção nessa cifra. A primeira delas é que o processo de cifragem e decifragem são iguais, um tipo de cifra chamada de involutiva. O segundo fato que chama a atenção é a ausência de chave. Veremos mais adiante o porquê de um método onde não há chave ser desaconselhável. Para exemplificar o uso desse método, a cifragem de “atbash hebraico” seria “ZGYZHS SVYIZRXL” e ao cifrarmos o texto “zgyzhs svyizrxl” obtemos de volta “ATBASH HEBRAICO”.

1.3 A CÍTALA ESPARTANA

Esse método, atribuído aos gregos, consiste em um bastão no qual enrolava-se uma tira de pergaminho, de maneira a cobrir o bastão todo e sem haver sobreposição do

pergaminho. A mensagem era quebrada em blocos de letras que eram escritos ao longo do eixo do cilindro, na vertical, cada letra em um setor diferente da mesma tira. Quando o pergaminho era desenrolado, o texto estava cifrado. Para decifrar a mensagem, o receptor precisava ter um outro bastão, de mesmo diâmetro, para enrolar a tira e ler a mensagem original.

Nesse método, não havia troca de letras, mas sim uma transposição na posição das letras. Exemplificando, vamos cifrar o texto "loucura isso e esparta". Escrevendo em colunas com 4 letras, teríamos:

l	u	s	e	r
o	r	s	s	t
u	a	o	p	a
c	i	e	a	

E lendo as linhas, ficaríamos com "LUSERORSSTUAOPACIEA".

1.4 A CIFRA DE VIGENÈRE

Essa cifra, atribuída a Blaise Vigenère (1523-1596). O melhor método para trabalhar com ele é relacionar cada letra do alfabeto com os números de 0 a 25 e operando com os números módulo 26.

A ideia dessa cifra é escolher uma chave, que pode ser uma palavra ou uma sequência qualquer de letras. Teoricamente, quanto maior o tamanho da chave (até o tamanho da mensagem a ser cifrada) mais difícil de decifrar o código. Aumentar o tamanho da chave além do tamanho da mensagem não influencia na segurança da cifra. O melhor jeito de entender o método é transformar as letras da chave e do texto original nos números de 0 a 25 e fazer a adição das letras do texto original com a chave, uma de cada vez.

Como exemplo, vamos escolher a chave k ="CRIPTO" para cifrar o texto m ="matemática". Fazendo a transformação das letras nos números, teremos como chave "2 17 8 15 19 14" e como texto "12 0 19 4 12 0 19 8 2 0". Para descobrir o texto cifrado, usamos $c_i = E_k(m_i) = (m_i + k_i) \pmod{26}$ temos:

Chave:	2	17	8	15	19	14	2	17	8	15
Texto:	12	0	19	4	12	0	19	8	2	0
Cifra:	14	17	1	19	5	14	21	25	10	15

E, transformando os números de volta em letras, temos “ORBTFOVZKP”. Na cifra de César, as letras iguais seriam transformadas sempre em letras iguais e isso nem sempre ocorre na cifra de Vigenère. Como exemplo, “matemática” possui a letra “a” três vezes, que foi transformada em “R”, “O” e “P”.

As transformações das letras eram realizadas por meio da *tabula recta*, uma tabela que indicava a letra a ser substituída de acordo com a letra da chave correspondente. Na figura abaixo, podemos ver um exemplo dela.

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

Figura 1: Tabula Recta

Com efeito, a cifra de César pode ser trabalhada como uma cifra de Vigenère, onde a chave tem tamanho 1.

O número de variações dessa cifra é abundante. Para os interessados em maiores informações, ver [6].

1.5 A CIFRA DE HILL

Essa cifra foi inventada em 1929 por Lester S. Hill. A ideia consiste em escolher um número inteiro positivo m e criar m transformações lineares para transformar m letras do texto original em m letras do texto cifrado.

Novamente, trabalharemos com o conjunto $\mathbb{Z}_{26}$. Se tivermos o texto $x = x_1x_2x_3\ldots x_n$ ele será transformado no texto cifrado $y = y_1y_2y_3\ldots y_n$. Se $n > m$, dividimos o texto x em quantos blocos de m letras forem necessários e juntamos os textos cifrados y ao final. A transformação será feita com as operações

$$\begin{bmatrix} y_1 & y_2 & \ldots & y_m \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & \ldots & x_m \end{bmatrix}.K$$

onde K é uma matriz quadrada inversível módulo 26 de ordem m. Precisamos que K seja inversível pois para decifrar faremos

$$\begin{bmatrix} x_1 & x_2 & \ldots & x_m \end{bmatrix} = \begin{bmatrix} y_1 & y_2 & \ldots & y_m \end{bmatrix} \cdot K^{-1}$$

Para exemplificar, pegaremos $m = 2$ e usaremos a chave $K = \begin{bmatrix} 3 & 5 \\ 4 & 13 \end{bmatrix}$ para cifrar o texto "café". Temos $x = (2, 0, 5, 4)$, logo

$$\begin{bmatrix} y_1 & y_2 \end{bmatrix} = \begin{bmatrix} 2 & 0 \end{bmatrix} \cdot \begin{bmatrix} 3 & 5 \\ 4 & 13 \end{bmatrix} = \begin{bmatrix} 6 & 10 \end{bmatrix}$$

$$\begin{bmatrix} y_3 & y_4 \end{bmatrix} = \begin{bmatrix} 5 & 4 \end{bmatrix} \cdot \begin{bmatrix} 3 & 5 \\ 4 & 13 \end{bmatrix} = \begin{bmatrix} 5 & 25 \end{bmatrix}$$

portanto $y = (6, 10, 5, 25)$, ou seja, o texto cifrado é "GKFZ". Para decifrar, usaríamos a matriz $K^{-1} = \begin{bmatrix} 13 & 23 \\ 8 & 7 \end{bmatrix}$.

A matriz inversa K^{-1} é encontrada resolvendo $K \cdot K^{-1} = I \pmod 2 6$, onde I é a matriz identidade de ordem m.

1.6 A MÁQUINA ENIGMA

Com o avanço tecnológico, foram criadas máquinas para cifrar e decifrar textos. Uma das mais famosas é a máquina Enigma, usada pelos alemães durante a Segunda Guerra Mundial.

Criada pelo engenheiro alemão Scherbius, é um mecanismo eletromecânico que consistia de um teclado de 26 letras e um painel luminoso com 26 letras. Ao apertar uma tecla, uma corrente elétrica percorria o mecanismo e fazia acender uma das letras no painel luminoso, que seria a letra cifrada. O caminho percorrido dependia da posição inicial dos discos que compunham a máquina, sendo difícil a quebra do código sem o conhecimento dessa posição inicial.

A máquina foi decifrada por criptoanalistas poloneses e ingleses, entre eles, notavelmente, estava Alan Turing (1912 - 1954), que por volta de 1943, em Bletchley Park, construíram dispositivos eletro-eletrônicos denominados Colossos para decodificar as mensagens alemãs. Essas máquinas seriam os percursores dos computadores modernos.

1.7 O PRINCÍPIO DE KERCKHOFF

Auguste Kerckhoff, em 1883, escreveu um tratado intitulado *La Cryptographie Militaire*. Nele discutiu princípios para cifras militares e um dos mais importantes desses princípios leva seu nome que diz:

O funcionamento interno de um criptossistema não pode ser secreto; deve-se presumir que o adversário conhece como o criptossistema funciona, e a segurança do sistema deve estar na escolha das chaves

Esse princípio é de fundamental importância para a criptografia moderna e diz que não podemos assumir que um sistema é seguro com base no método de cifragem, ele deve manter-se seguro mesmo se o algoritmo for conhecido por todos. Existem diversos argumentos a favor do princípio de Kerckhoff.

Em primeiro lugar, é muito mais fácil manter em segredo uma chave relativamente pequena que um algoritmo. Na computação moderna, uma cadeia de bits é mais fácil de armazenar e compartilhar em segredo que um programa ou algoritmo, muitas vezes maior. Além disso, detalhes sobre o algoritmo podem se tornar conhecidos, talvez

por alguém que ajudou a criar o método divulgando a informação ou por engenharia reversa.

Outro ponto em favor do princípio é que, caso uma chave seja exposta, é muito mais fácil trocar a chave do que trocar o algoritmo sendo usado. Existem vários pontos em favor do algoritmo ser necessariamente público para melhorar a segurança do criptossistema [8].

Além desses pontos, um algoritmo público pode ter sua segurança verificada por diversas pessoas, sendo possível ter uma maior confiança no sistema do que se a validação fosse feita por um número pequeno de pessoas.

Um caso famoso que exemplifica a importância desse princípio é o do protocolo GSM, usado por grande parte dos celulares atuais. O design da criptografia desse algoritmo não foi publicado e isso não é suficiente para manter sua segurança. Na verdade, há numerosos casos de clonagem e até algoritmos para quebra de mensagens [1].

1.8 SIGILO PERFEITO

Em 1949, Claude Shannon (1916-2001) publica seu artigo *Communication Theory of Secrecy Systems*. Nesse artigo, Shannon discute a segurança de mensagens e introduz o conceito de *sigilo perfeito*.

Uma cifra tem *sigilo perfeito* se o conhecimento do texto cifrado não fornece informação alguma sobre o texto original [11], ou seja, ele resiste a todos os ataques baseados apenas no texto cifrado, mesmo que o observador possua recursos computacionais e tempo infinitos.

Mais precisamente, sendo $Pr(m)$ a probabilidade que a mensagem enviada seja m e $Pr(m|c)$ a probabilidade condicional de que a mensagem seja m sabendo-se que o texto criptografado é c, isso é o mesmo que dizer

$$Pr(m|c) = Pr(m).$$

Ou seja, a probabilidade de um texto qualquer ser m é a mesma não importando qual o texto encriptado.

Um exemplo de criptografia perfeitamente segura é o *one-time pad de Vernam* [5].

1.9 ONE-TIME PAD

O one-time pad tem um funcionamento similar à cifra de Vigenère, mas a chave precisa ser completamente aleatória e ter tamanho igual ao da mensagem. Para exemplificar, vamos usar como mensagem a frase "hoje tem café", uma frase com 11 caracteres (desconsiderando os espaços). Usaremos como chave uma sequência aleatória de 11 letras, por exemplo "wnfqspotfph". Fazendo a adição módulo 26, temos:

Mensagem 1:	h o j e t e m c a f e	7	14	9	4	19	4	12	2	0	5	4
Chave 1:	w n f q s p o t f p h	22	13	5	16	18	15	14	19	5	15	7
Cifra:	D B O U L T A V F U L	3	1	14	20	11	19	0	21	5	20	11

Como a chave é completamente aleatória, alguém que intercepte o texto cifrado "D B O U L T A V F U L" só tem como opção "chutar" uma chave e esperar que o resultado dê uma mensagem coerente. Digamos que a chave testada seja "pzophtyvegr". Fazendo a subtração módulo 26, temos:

Cifra:	D B O U L T A V F U L	3	1	14	20	11	19	0	21	5	20	11
Chave 2:	p z o p h t y v e g r	15	25	14	15	7	19	24	21	4	6	17
Mensagem 2:	o c a f e a c a b o u	14	2	0	5	4	0	2	0	1	14	20

A mensagem alternativa que obtemos é "o café acabou", que faz sentido o suficiente mas que possui significado completamente oposto ao original. Qualquer mensagem original de 11 letras possui uma chave (única) tal que o texto cifrado será "D B O U L T A V F U L", pois a chave é aleatória e poderia ser qualquer sequência, logo qualquer conjunto de 11 caracteres pode ter gerado essa cifra e o observador não ganha informação alguma estudando o texto cifrado.

Outra maneira de trabalhar o one-time pad é com bits e a operação fica muito mais fácil de ser efetuada. Suponha uma mensagem m com n bits e uma chave k, aleatória e também com n bits. O texto cifrado c é obtido por $c_i = m_i \oplus k_i$, sendo i a posição do bit, com $i = 1, 2, ..., n$ e $\oplus$ representando a operação ou exclusivo, ou seja, realizando a operação ou exclusivo bit a bit entre a mensagem e a chave. Note que para realizar a inversa, basta aplicar o mesmo algoritmo novamente.

Para exemplificar, tomemos $n = 8$ e a mensagem $m = 01100010$. Com a chave $k = 10010011$, temos como texto cifrado $c = m \oplus k = 11110001$. Para termos a mensagem

original, basta fazer $m = c \oplus k = 01100010$. É possível demonstrar que o one-time pad tem sigilo perfeito, o que será feito na proposição abaixo.

Proposição 1. O one-time pad tem sigilo perfeito.

Demonstração. Seja C um bit qualquer da chave. Por hipótese, a chave do one-time pad é completamente aleatória, logo

$$Pr(C = 0) = Pr(C = 1) = \frac{1}{2}$$

Considere uma mensagem onde M seja um bit qualquer da mensagem, temos que

$$Pr(M = 0) = 1 - Pr(M = 1) = x$$

Ao calcularmos o bit do texto cifrado $B = M \oplus C$, temos como probabilidade de B as seguintes opções:

$$Pr(B = 0) = Pr(M = 0) \cdot Pr(C = 0) + Pr(M = 1) \cdot Pr(C = 1) = \frac{x}{2} + \frac{1 - x}{2} = \frac{1}{2}$$

$$Pr(B = 1) = Pr(M = 0) \cdot Pr(C = 1) + Pr(M = 1) \cdot Pr(C = 0) = \frac{x}{2} + \frac{1 - x}{2} = \frac{1}{2}$$

Ou seja, o bit B do texto cifrado é completamente aleatório. □

A demonstração dessa proposição no conjunto $\mathbb{Z}_{26}$, que corresponde ao nosso alfabeto, é análoga, mas a probabilidade de cada letra será $\frac{1}{26}$.

Infelizmente, há grandes desvantagens em usar o one-time pad, entre elas a chave que precisa ser de tamanho igual ao da mensagem, aleatória e não reutilizada e que precisa ser comunicada por um canal completamente seguro [12]. Logicamente que, se um canal completamente seguro para passar a chave estivesse disponível, seria muito mais fácil passar a mensagem por ele, pois tanto a chave quanto a mensagem possuem o mesmo tamanho.

Nesse ponto, existem vantagens em trabalhar com sistemas que não sejam perfeitamente seguros, mas que mesmo assim possam ter sua segurança demonstrada.

1.10 SEGURANÇA DEMONSTRÁVEL

Na Criptografia Moderna usamos a noção de *segurança demonstrável*, que significa que um sistema criptográfico qualquer deve ter sua segurança demonstrada rigoro-

samente, pressupondo hipóteses de segurança e enunciando claramente conclusões, como uma demonstração de teorema.

No contexto da segurança demonstrável, buscam-se provas rigorosas que assegurem que um criptossistema resiste a certos tipos de ataque [5]. Para saber se um sistema é demonstravelmente seguro, devem ser feitas considerações reais sobre as capacidades do oponente, considerando-se somente ataques possíveis. Normalmente, são usados algoritmos probabilísticos para modelar os atacantes e os tipos de ataques a serem utilizados.

A segurança de um sistema é baseada em problemas computacionais de difícil solução. Por exemplo, se computar os fatores primos de um número inteiro fosse um problema de fácil solução (se houvesse um algoritmo de tempo polinomial que o fizesse), seria relativamente fácil descobrir as chaves do sistema RSA. A prova que um sistema de chave pública é seguro normalmente depende do fato de uma certa função ser de mão única [5], que será explicado no próximo capítulo.

2

FUNÇÕES DE MÃO ÚNICA

Estudaremos nesse capítulo algumas bases necessárias para o desenvolvimento do trabalho, definindo as funções de mão única e fornecendo algumas bases teóricas para seu uso.

Primeiramente, começamos definindo o que é uma função desprezível.

2.1 FUNÇÃO DESPREZÍVEL

Definição 2.1 (Função Desprezível)**.** Uma função f é desprezível se, para qualquer polinômio $p(.)$ existe N tal que para todos os inteiros $n > N$ temos $f(n) < \frac{1}{p(n)}$.

Podemos dizer que uma função é desprezível se ela cresce mais lentamente que o recíproco de qualquer polinômio.

Como, para qualquer polinômio $p(x)$ de grau n existe um monômio x^c, $c > n$, que cresce mais rapidamente, para mostrar que uma função f é desprezível, basta provar que para cada c existe N tal que para todos os inteiros $n > N$ temos $f(n) < \frac{1}{n^c}$.

É importante notar que $f(n) = \frac{1}{2^n}$ é uma função desprezível. A importância desse resultado é devido ao fato de, ao se trabalhar com um alfabeto binário, contendo apenas $0, 1$, a quantidade de palavras possíveis com n caracteres é 2^n, logo a probabilidade de se escolher uma em particular é $\frac{1}{2^n}$, sendo essa a probabilidade de descobrir a chave ao selecionar um valor aleatoriamente.

Para manter coerência com a literatura, utilizaremos a notação em inglês para função desprezível, **negl**.

A seguir, encontram-se duas proposições interessantes sobre as funções desprezíveis.

Teorema 2.2. *Sejam f e g duas funções desprezíveis. Então $f+g$ é uma função desprezível.*

Demonstração. Se f e g são desprezíveis, então existem $N_1, N_2 \in \mathbb{Z}$ tal que $\forall n_1 > N_1, f(n_1) < \frac{1}{n_1^{c_1}}$ e $\forall n_2 > N_2, g(n_2) < \frac{1}{n_2^{c_2}}$.

Seja $N = \max(2, N_1, N_2)$ e $c = \max(c_1, c_2)$, então, para $n > N$, temos

$$\begin{aligned} f(n)+g(n) &< \frac{1}{n^c} + \frac{1}{n^c} = \\ &= \frac{2}{n^c} \leq \\ &\leq \frac{n}{n^c} = \\ &= \frac{1}{n^{c-1}} \\ \therefore f(n)+g(n) &< \frac{1}{n^{c-1}} \quad \square \end{aligned}$$

Teorema 2.3. *Sejam $p(.)$ um polinômio positivo qualquer e f uma função desprezível. Então, o produto $p \cdot f$ é uma função desprezível.*

Demonstração. Seja x^c um monômio de grau c qualquer, logo $x^c \cdot p(x)$ é um polinômio. Como f é uma função desprezível, temos que existe N tal que $\forall n > N$,

$$f(n) < \frac{1}{n^c \cdot p(n)} \implies p(n) \cdot f(n) < \frac{1}{n^c}$$

$\therefore p.f$ é uma função desprezível $\square$

2.2 FUNÇÃO DE MÃO ÚNICA

Uma função de mão única é uma espécie de função fácil de calcular mas cuja inversa é difícil de ser calculada.

Definição 2.4 (Função de Mão Única)**.** Uma função f é dita de mão única se ela possui os seguintes requisitos:

1. Fácil de calcular: Existe um algoritmo que calcule f em tempo polinomial;
2. Difícil de inverter: Qualquer algoritmo aleatório A de tempo polinomial deve ter probabilidade desprezível de encontrar x dado $f(x)$, ou seja, $\forall p(n)$,

$$\Pr\left[A(f(x)) \in f^{-1}(f(x))\right] < \frac{1}{p(n)}$$

Infelizmente, não sabemos se tais funções existem. Aquelas que consideramos serem boas candidatas para serem funções de mão única, se baseiam em conjecturas. [5]

Exemplos dessas funções são a exponenciação modular, que se baseia na conjectura de que não existe maneira eficiente de se calcular o logaritmo discreto, e a soma de subconjuntos, cuja inversa é um problema $\mathcal{NP}$-completo e portanto depende da conjectura que $\mathcal{NP} \neq \mathcal{P}$.

Estudaremos agora algumas funções que são candidatas a serem funções de mão única.

2.2.1 *Exponenciação Discreta*

Para falar da exponenciação discreta, precisamos primeiramente definir o que é uma raiz primitiva módulo n. Dizemos que g é uma raiz primitiva módulo n se, para cada inteiro a coprimo com n, existe um inteiro k tal que $g^k = a \pmod{n}$. Podemos dizer também que g é um gerador do grupo multiplicativo dos inteiros módulo n.

Seja p um número primo e g uma raiz primitiva de $\mathbb{Z}_p$. A função exponenciação discreta é definida por

$$\begin{aligned} \operatorname{dexp} : \mathbb{Z}_p^* &\to \mathbb{Z}_p^* \\ x &\mapsto \operatorname{dexp}(x) = g^x \pmod{p} \end{aligned}$$

Essa função pode ser calculada em tempo polinomial [5] mas todos os algoritmos que se conhecem para calcular x dado $\operatorname{dexp}(x)$ exigem uma quantidade de operações que aumenta exponencialmente com o tamanho de x [5]. Realmente, conjectura-se que não exista função que calcule a inversa de dessa função - esse problema é conhecido como “problema do logaritmo discreto”.

É interessante notar que essa função também é uma bijeção, visto que g é uma raiz primitiva de $\mathbb{Z}_p$ e $\mathbb{Z}_p^*$ é um sistema reduzido de resíduos módulo p, logo a inversa sempre existe e é única - ainda que difícil de ser calculada.

2.2.2 *Multiplicação de Inteiros*

A função multiplicação de inteiros é definida por

$$\cdot : \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}$$

$$x, y \mapsto x \cdot y$$

A multiplicação de números inteiros é uma função relativamente fácil de realizar, existindo vários algoritmos que a calculam em tempo polinomial. Se não houver restrições, ela também pode ser invertida de maneira relativamente fácil, pois há uma grande probabilidade que $x \cdot y$ tenha um fator primo p pequeno que possa ser encontrado e então $(p, \frac{xy}{p})$ é uma pré imagem de $x \cdot y$. Mas, ao restringir x e y de maneira que ambos sejam números primos, grandes e de tamanhos aproximadamente iguais, a função multiplicação de inteiros torna-se uma função que conjectura-se ser de mão única [8].

De fato, não se conhece algoritmo que descubra x e y dado $x \cdot y$ em tempo polinomial, se x e y forem primos.

2.2.3 *Quadrado Modular*

Seja a função

$$Sq_n : \mathbb{Z}_n^* \to \mathbb{Z}_n^*$$

$$x \mapsto x^2$$

Essa função não é injetora e nem sobrejetora. Por exemplo, em $\mathbb{Z}_3$, temos

$$0^2 = 0 \pmod 3$$

$$1^2 = 1 \pmod 3$$

$$2^2 = 1 \pmod 3$$

Calcular $x^2 \pmod n$ pode ser feito em tempo polinomial e existem algoritmos probabilísticos polinomiais que calculam a raiz quadrada módulo p, se p for um número primo. Mas, se $n = pq$ e a fatoração de n é conhecida, é relativamente fácil calcular a raiz quadrada módulo n pelo Teorema Chinês do Resto.

Na verdade, o problema de calcular a raiz quadrada módulo n é análogo ao de fatorar $n = pq$, se p e q forem primos [5].

2.2.4 *Soma de Subconjuntos*

Sejam n inteiros $x_1, x_2, \ldots, x_n$ e um subconjunto S deles, temos a função

$$\textbf{somaS}(x_1, x_2, \ldots, x_n, S) = (x_1, x_2, \ldots, x_n, \sum_{i \in S} x_i)$$

Essa função é fácil de calcular, mas descobrir o subconjunto S a partir da soma é um problema $\mathcal{NP}$-completo e não se conhecem algoritmos eficientes para a resolução desse problema [8].

2.3 PREDICADO HARD-CORE

Como dito anteriormente, em uma função de mão única f, deve ser difícil calcular a pré-imagem x dado $y = f(x)$, mas algumas propriedades de x podem ser obtidas com facilidade. Um dos exemplos mais conhecidos disso ocorre na exponenciação discreta, no qual a paridade de x pode ser calculada usando o critério de Euler [5].

Um predicado hard-core de x relativo a f é uma das propriedades da pré-imagem x que é muito difícil de ser descoberta dada a imagem $f(x)$.

Definição 2.5 (Predicado Hard-Core)**.** $H : \{0,1\}^* \to \{0,1\}$ é um predicado hard-core de uma função f se, para cada algoritmo probabilístico em tempo polinomial $\mathcal{A}$, existe uma função desprezível **negl** tal que

$$Pr[\mathcal{A}(f(x)) = H(x)] \leq \frac{1}{2} + \textbf{negl}(n)$$

onde n é o número de bits de x.

3

GERADORES DE NÚMEROS PSEUDO-ALEATÓRIOS

A geração de números aleatórios é um assunto importante no ramo da criptografia, com diversas aplicações - como o one-time pad mencionado anteriormente. Mas a verdadeira aleatoriedade é difícil de se encontrar, se é que pode-se encontrá-la de verdade. Usando computadores, podemos gerar grandes sequências numéricas usando fórmulas e outros métodos aritméticos, mas esses números não são realmente aleatórios. Como disse o matemático Von Neumann, "qualquer um que pense em métodos aritméticos para produzir números aleatórios está, obviamente, cometendo pecados" [14].

Por isso usamos sequências numéricas que são pseudo-aleatórias, isso é, são sequências de números que podem ser consideradas como se fossem geradas por um evento aleatório, como o lançar de uma moeda ou de um dado.

Idealmente, um gerador pseudo-aleatório deve pegar uma sequência s aleatória de tamanho n e retornar uma sequência $G(s)$ de tamanho $l(n)$. Para considerar G um gerador de números pseudo-aleatórios, usaremos a seguinte definição:

Definição 3.1 (Gerador Pseudo-Aleatório)**.** Seja l um polinômio e G um algoritmo determinístico de tempo polinomial que com uma entrada s, com $s \in \{0,1\}^n$, retorne $G(s)$ de tamanho $l(n)$. G é um gerador pseudo-aleatório se:

1. $\forall n, l(n) > n$. Dizemos que l é o fator de expansão do gerador.
2. Para qualquer função probabilística de tempo polinomial que verifique se um número vem de um sequência aleatória uniforme D, existe uma função desprezível **negl** tal que $|P(D(r) = 1) - P(D(G(s) = 1)| \leq$ **negl**(n), onde r é um número de tamanho $l(n)$ escolhido aleatoriamente e s é a *semente* escolhida aleatoriamente e de tamanho n.

Um dos principais problemas dos geradores pseudo-aleatórios é que, da mesma forma das funções de mão única, não se sabe se eles existem de fato [8]. De fato, a maioria dos geradores existentes baseia-se em funções de mão única [13], inclusive um dos mais populares,o Gerador de Blum-Blum-Shub [2], que veremos adiante.

Cabe ressaltar que, se um gerador pseudo-aleatório é indistinguível em tempo polinomial de uma sequência aleatória, como definido acima, ele pode ser usado para criptografar de maneira demonstravelmente segura usando o princípio do one-time pad visto em 1.9 com chave de segurança s, pois não é possível para um observador distinguir $G(s)$ de uma sequência aleatória em tempo polinomial [8].

De maneira geral, um gerador de números pseudo-aleatório pode ser construído de maneira segura usando uma função de mão única com um predicado hard-core [5] [3].

3.1 GERADOR DE NÚMEROS PSEUDO-ALEATÓRIOS DE BLUM-BLUM-SHUB

Esse gerador, também chamado de Gerador $x^2 \pmod N$ usa a operação de quadrado modular que conjecturamos ser uma função de mão única anteriormente em 2.2.3. Para usá-lo, precisamos de dois números primos p e q, com $p \neq q$ e $p = q = 3 \pmod 4$. Calculamos então $N = p.q$. É importante que p e q sejam de mesmo tamanho, para dificultar a fatoração de N.

Como semente para esse gerador, tomamos um número aleatório s, com $s \neq 0 \pmod N$ e $s^2 \neq 1 \pmod N$. Para que s^2 tenha essa restrição, obviamente temos que $s \neq \pm 1 \pmod N$, mas também é necessário que $s \neq \pm 1 \pmod p$ e $s \neq \pm 1 \pmod q$.

Esse gerador calcula $G(s) = b_1 b_2 b_3 ... b_n$ onde

$$\begin{cases} s_0 = s \\ s_i = s_{i-1}^2 \pmod N \\ b_i = s_i \pmod 2, i = 1, 2, \ldots, n \end{cases}$$

Para usar um exemplo, tomemos $p = 83$ e $q = 79$, ou seja, $N = 6557$. Como semente, usaremos $s = 42$. A tabela abaixo ilustra o valor de m_i e b_i com $0 \leq i \leq 16$.

i	s_i	b_i
0	42	
1	1764	0
2	3678	0
3	593	1
4	4128	0
5	5298	0
6	4844	0
7	3390	0
8	4236	0
9	3744	0
10	5227	1
11	5067	1
12	3834	0
13	5319	1
14	4863	1
15	4227	1
16	6261	1

Logo, $G(42) = 0010000001101111\ldots$

É importante notar que após um certo valor n, começa a haver uma repetição nos dígitos de $G(s)$, ou seja, G é periódica. É possível calcular o período de $G(s)$ [2], mas isso não sera abordado nesse trabalho.

4

FUNÇÕES DE HASH

As funções de hash, também conhecidas como resumos criptográficos, são usadas para transformar cadeias longas de informação em um valor de tamanho fixo, usualmente menor que a informação inicial. Um dos principais usos para essas funções é o de checar a integridade de mensagens, chaves e arquivos. Nesse sentido, a função de hash funciona como uma impressão digital da informação. Idealmente, se a mensagem for alterada, mesmo que somente em um bit, a impressão digital também deverá ser alterada de uma maneira "difícil de ser prevista".

Seja m uma mensagem arbitrariamente longa e h uma função de hash. A imagem $h(m)$ terá sempre um tamanho fixo n. A escolha desse tamanho n será importante para determinar a segurança da função, como veremos adiante. Atualmente, costuma-se usar o tamanho n entre 128 e 512 bits [5]. Como a função de hash possui um domínio infinito e um contradomínio finito, certamente haverá um número (infinito) de mensagens diferentes que possuirão a mesma imagem. Na prática, o domínio de uma função de hash não é infinito, pois há um limite no tamanho das mensagens a serem enviadas, mas possui um número de elementos arbitrariamente grande.

Uma função de hash deve cumprir alguns requisitos de segurança para ser eficiente.

4.1 REQUISITOS DAS FUNÇÕES DE HASH

Para que uma função de hash seja segura, é necessário que alguns problemas sejam de difícil solução.

A princípio, queremos que a mensagem m, dado o resumo $h(m)$, seja difícil de ser encontrada. Podemos dizer também que há probabilidade desprezível de um algoritmo

que funcione em tempo polinomial de, dado uma imagem y da função de hash, encontrar m tal que $h(m) = y$. Se esse requisito for cumprido, diz-se que a função de hash possui **resistência de pré-imagem**, ou ainda, que ela é de **mão única**.

Outro problema a ser tratado já foi mencionado anteriormente, o de que há mensagens $m \neq m'$ tal que $h(m) = h(m')$. Se, dado m, encontrar $m' \neq m$ tal que $h(m') = h(m)$ for um problema de difícil solução, dizemos que a função de hash possui **resistência de segunda pré-imagem**.

Por último, se, em uma função de hash h qualquer, for difícil encontrar duas mensagens m e m', com $m \neq m'$ e $h(m) = h(m')$, a função de hash será **resistente à colisão**. Essas três propriedades são essenciais para a segurança de uma função de hash.

É possível mostrar que a resistência a colisão engloba a resistência à segunda pré-imagem e a resistência à pré-imagem, como faremos a seguir.

Proposição 4.1. *Uma função de hash que tenha resistência à segunda pré-imagem também é resistente à pré-imagem.*

Demonstração. Suponha que a função de hash h não possua resistência a pré-imagem, ou seja, dada uma imagem y da função, há uma probabilidade não-desprezível de se encontrar algum m tal que $h(m) = y$.

Como já foi dito antes, a função de hash possui um número infinito de mensagens diferentes que possuirão a mesma imagem.

Dado uma mensagem m qualquer, calcula-se $y = h(m)$. Como a função de hash h não possui resistência a pré-imagem, há uma grande probabilidade de se calcular m' onde $y = h(m')$, mas como existem um número infinito de mensagens que possuem imagem y, a probabilidade de que $m \neq m'$ é alta, logo foi possível encontrar $m' \neq m$ dado m tal que $h(m) = h(m')$, ou seja, a função de hash h não possui resistência a segunda pré-imagem, o que demonstra a proposição por contraposição. □

Proposição 4.2. *Uma função de hash que tenha resistência à colisão também é resistente à segunda pré-imagem.*

Demonstração. Novamente, por contraposição, suponha que a função de hash h não possua resistência à segunda pré-imagem, ou seja, dado uma mensagem m há uma probabilidade não-desprezível de se encontrar $m' \neq m$ tal que $h(m) = h(m')$.

Seja m uma mensagem aleatória qualquer. Como a função de hash h não possui resistência à segunda pré-imagem, é possível encontrar $m' \neq m$ tal que $h(m) = h(m')$,

logo é possível encontrar duas mensagens m e m', com $m \neq m'$ onde $h(m) = h(m')$, logo a função de hash h não possui resistência à colisão. □

Com essas duas proposições, podemos concluir que a resistência à colisão é uma condição de segurança mais forte que as outras duas anteriores, pois as engloba.

4.2 PROBLEMA DO ANIVERSÁRIO

O problema do aniversário é um problema clássico em probabilidade. Em um grupo de n pessoas, qual a probabilidade $p(n)$ de que ao menos duas delas façam aniversário no mesmo dia?

Como um ano possui 365 dias, precisamos resolver o problema para $n \leq 365$, pois se $n > 365$, pelo Princípio da Casa dos Pombos, sempre haverá duas pessoas que nasceram no mesmo dia.

É relativamente fácil calcular a probabilidade de que todos os n aniversários sejam diferentes, o complementar de $p(n)$, que será

$$\bar{p}(n) = \frac{365}{365} \cdot \frac{364}{365} \cdots \frac{365 - n + 1}{365} = \frac{365!}{365^n \cdot (365 - n)!}$$

logo $p(n)$ é

$$p(n) = 1 - \frac{365!}{365^n \cdot (365 - n)!}$$

Com essa fórmula, podemos perceber que, em um grupo de 23 pessoas, há uma probabilidade de aproximadamente $50,7\%$ de que ao menos duas possuam a mesma data de aniversário e que em um grupo de 50 pessoas, essa probabilidade chega a 97%.

Agora, imagine uma função de hash onde N é o número de elementos do conjunto imagem. Qual a probabilidade $p(n)$ de que, ao calcular o hash de n mensagens diferentes, elas possuam a mesma imagem? Do jeito que foi enunciado, é possível perceber que esse problema é um caso genérico do problema do aniversário, mas ao invés de termos 365 dias de aniversário diferentes, possuímos N resumos diferentes. Essa probabilidade será calculada por

$$p(n) = 1 - \frac{N!}{N^n \cdot (N - n)!}$$

Esse resultado leva ao chamado "ataque do aniversário", pois é possível deduzir que se $n \geq \frac{\sqrt{1+8\ln 2 \cdot N}+1}{2} \approx 1,18\sqrt{N}$ [5] a probabilidade de encontrar ao menos uma colisão é de $\frac{1}{2}$.

4.3 CONSTRUÇÃO DE FUNÇÃO DE HASH

O exemplo a seguir descreve uma função de hash devida a Chaum, van Heijst e Pfitzmann [4], também chamada de função de hash do logaritmo discreto. Apesar de ser muito lenta para ser usada na prática, ela ilustra como uma função de hash funciona.

É necessário escolher um número primo grande p tal que $q = \frac{p-1}{2}$ também seja primo. Em seguida, escolhemos duas raízes primitivas de p, sejam elas α e β.

Em seguida, seja m a mensagem a ser resumida, escrevemos $m = x_0 + x_1 q$, com $0 \leq x_0, x_1 \leq q - 1$. A função de hash h é definida por $h(m) = \alpha^{x_0}\beta^{x_1} \pmod{p}$. É importante notar que a mensagem m deve ser um menor que q^2 devido às restrições dadas a x_1 anteriormente.

Como exemplo, vamos escolher $p = 503$, pois $q = \frac{503-1}{2} = 251$ é também um número primo. Como raízes primitivas de p, escolhemos $\alpha = 10$ e $\beta = 30$ como raízes primitivas de 503. Como mensagem, escolhemos $m = 5200$.

Temos que $5200 = 180 + 20 \cdot 251$, logo $x_0 = 180$ e $x_1 = 20$, então $h(5200) = 10^{1}80 \cdot 30^{2}0 \pmod{503}$ e efetuando os cálculos teremos $h(5200) = 336$

Um método bem usado para construir uma função de hash é a construção de Merkle-Damgård. Esse método transforma uma função de compressão, um tipo de função de mão única, em uma função de hash que obedecerá a especificações de segurança como resistência à colisão se a função de compressão também a possuir. [12]

4.4 RESÍDUOS QUADRÁTICOS COMO FUNÇÃO DE HASH

Essa construção usa como ideia básica o gerador de números pseudo-aleatórios de Blum-Blum-Shub [2].

Para isso, é necessário encontrar um número $N = p \cdot q$, onde $p = q = 3 \pmod 4$ e $p \neq q$. A mensagem m deve ter um tamanho fixo, $m < N$. Se $m \geq N$, podemos usar a construção de Merkle-Damgård para dividir m de maneira a termos um tamanho adequado.

Definimos $f(m) = b_0 b_1 b_2 \dots b_n$, onde b_i é a paridade de m_i e $m_{i+1} = m_i^2 \pmod N$, com $m = m_0$. O valor de n, o tamanho em bits da imagem de f é arbitrário e nesse exemplo vamos fixá-lo como $n = 16$.

Por exemplo, tomemos $p = 83$ e $q = 79$, logo $N = 6557$. Vamos pegar como mensagem $m = 123$. A tabela abaixo ilustra o valor de m_i e b_i com $0 \leq i \leq 15$.

i	m_i	$m_i \pmod N$	b_i
0	123	123	1
1	15129	2015	1
2	4060225	1442	0
3	2079364	795	1
4	632025	2553	1
5	6517809	151	1
6	22801	3130	0
7	9796900	742	0
8	550564	6333	1
9	40106889	4277	1
10	18292729	5256	0
11	27625536	895	1
12	801025	1071	1
13	1147041	6123	1
14	37491129	4760	0
15	22657600	3165	1

Logo $f(123) = 1101110011011101$. A escolha dos primos e da mensagem tem apenas função de exemplificar, os primos usados na prática são muito maiores, assim como o tamanho da imagem f.

Para encontrar uma segunda pré-imagem da mensagem, seria necessário obter um outro elemento $x \neq 123$ tal que $f(x) = 1101110011011101$. Como visto anteriormente, temos chance de $\frac{1}{2}$ de achar uma segunda pré-imagem ao testar cerca de $1,18\sqrt{N} \approx 96$, o que pode ser obtido rapidamente devido ao pequeno valor de N.

5

ENCRIPTAÇÃO COM CHAVE PÚBLICA

Em 1976, Whitfield Diffie e Martin Hellman publicam o artigo "New Directions in Cryptography", onde sugerem as bases do método criptográfico de chave pública, de grande utilidade nos tempos de internet.

Imagine que Alice e Bob querem trocar uma mensagem, mas eles estão distantes e não há um canal seguro através do qual eles possam concordar em uma chave. Alice então escolhe uma chave pública pk e uma chave secreta sk. Qualquer um que queira comunicar-se com ela, utiliza a chave pública ao fazer a encriptação e apenas Alice deve ser capaz de decifrar a mensagem usando sua chave secreta.

Para compreender o processo, podemos pensar na chave pública como um cadeado aberto e a chave secreta como a chave que abre o cadeado. Qualquer um pode pegar o cadeado aberto e travar a mensagem com ele, mas apenas Alice que tem a chave secreta correspondente a esse cadeado pode destravá-lo.

Antes de 1976, James Ellis já havia proposto sobre a criptografia de chave pública em seu artigo intitulado "The possibility of non-secret encryption", mas esse documento era segredo do governo britânico e só foi liberado em 1997 [12].

Em 1977, Ronald Rivest, Adi Shamir e Leonard Adleman inventaram o criptosistema RSA, um dos mais famosos criptossistemas de chave pública, usado ainda nos dias de hoje.

Existem vários métodos de encriptação com chave pública. Trataremos agora sobre o criptossistema de Rabin.

5.1 CRIPTOSSISTEMA DE RABIN

O criptossistema de Rabin tem como base os resíduos quadráticos e tem segurança demonstrável, sendo tão difícil de decifrar quanto a fatoração de grandes números [10].

Ele foi criado por Michael Rabin em 1979 e é considerado um dos primeiros criptossistemas de chave pública. O processo de encriptação é relativamente simples, como será explicado a seguir.

Sejam $n = p \cdot q$, onde p e q são dois números primos tais que $p = q = 3 \pmod 4$, e m uma mensagem tal que $m < n$. A encriptação consiste em calcular y tal que $y = m^2 \pmod n$. A chave pública nesse sistema é n, o que permite a qualquer um que escreva a mensagem calcular y, mas só quem possui a chave privada p, q pode descriptografar.

Para decifrar a mensagem, precisamos lembrar do Critério de Euler, que por ser um resultado muito conhecido aceitaremos sem demonstração:

Critério de Euler. Se p é um número primo ímpar e $a \in \mathbb{Z}$ é coprimo com p, então:

- $a^{\frac{p-1}{2}} = 1 \pmod p$ se, e somente se, a é resíduo quadrático módulo p.
- $a^{\frac{p-1}{2}} = -1 \pmod p$ se, e somente se, a não é resíduo quadrático módulo p.

Como y é um resíduo quadrático módulo n, e portanto resíduo quadrático módulo p e módulo q, temos:

$$\begin{aligned} 1 &= y^{\frac{p-1}{2}} \pmod p \\ y &= y \cdot y^{\frac{p-1}{2}} \pmod p \\ &= y^{\frac{p-1}{2}+1} \pmod p \\ &= y^{\frac{p+1}{2}} \pmod p \\ &= (y^{\frac{p+1}{4}})^2 \pmod p \end{aligned}$$

Como $p = 3 \pmod 4$, $\frac{p+1}{4}$ é um número inteiro, logo

$$\pm\sqrt{y} = \pm y^{\frac{p+1}{4}} \pmod p$$

Analogamente, temos

$$\pm\sqrt{y} = \pm y^{\frac{q+1}{4}} \pmod q$$

Para encontrar a mensagem m, precisamos resolver o sistema

$$\begin{cases} m = \pm y^{\frac{p+1}{4}} \pmod{p} \\ m = \pm y^{\frac{q+1}{4}} \pmod{q} \end{cases}$$

As soluções desse sistema podem ser obtidas facilmente pelo Teorema Chinês dos Restos:

$$\begin{cases} m = q \cdot y^{\frac{p+1}{4}} \cdot a + p \cdot y^{\frac{q+1}{4}} \cdot b \pmod{n} \\ m = -q \cdot y^{\frac{p+1}{4}} \cdot a + p \cdot y^{\frac{q+1}{4}} \cdot b \pmod{n} \\ m = q \cdot y^{\frac{p+1}{4}} \cdot a - p \cdot y^{\frac{q+1}{4}} \cdot b \pmod{n} \\ m = -q \cdot y^{\frac{p+1}{4}} \cdot a - p \cdot y^{\frac{q+1}{4}} \cdot b \pmod{n} \end{cases}$$

Onde a e b são tais que $a \cdot q = 1 \pmod{p}$ e $b \cdot p = 1 \pmod{q}$

A desvantagem desse criptossistema é que ao decifrar a mensagem podemos ter quatro soluções (teremos duas soluções apenas se $p|m$ ou $q|m$) e, a não ser que o contexto diga qual das soluções é a correta, mais informação deve ser transmitida de maneira a não deixar dúvidas quanto a mensagem original. O motivo de ter quatro soluções diferentes é que, sendo ω tal que $\omega^2 = 1 \pmod{n}$ e $\omega \neq \pm 1 \pmod{n}$, temos que

$$m^2 = (-m)^2 = (\omega m)^2 = (-\omega m)^2 \pmod{n}$$

Como exemplo, tomemos como números primos $p = 7$ e $q = 11$ e a mensagem original é $m = 48$. Criptografando com a chave pública $n = 77$, temos $y = 48^2 \pmod{77} = 71$. Para decifrar a mensagem, temos que $a = 2$ e $b = 8$, logo as mensagens possíveis são:

$$\begin{cases} m_1 = 11 \cdot 48^{\frac{7+1}{4}} \cdot 2 + 7 \cdot 48^{\frac{11+1}{4}} \cdot 8 \pmod{77} = 15 \\ m_2 = -11 \cdot 48^{\frac{7+1}{4}} \cdot 2 + 7 \cdot 48^{\frac{11+1}{4}} \cdot 8 \pmod{77} = 48 \\ m_3 = 11 \cdot 48^{\frac{7+1}{4}} \cdot 2 - 7 \cdot 48^{\frac{11+1}{4}} \cdot 8 \pmod{77} = 29 \\ m_4 = -11 \cdot 48^{\frac{7+1}{4}} \cdot 2 - 7 \cdot 48^{\frac{11+1}{4}} \cdot 8 \pmod{77} = 62 \end{cases}$$

Apenas o contexto ou alguma outra informação deixaria claro qual dos quatro valores é a mensagem original.

É importante notar que fixamos p e q como sendo da forma $4k+3$ por existir um método determinístico de encontrar $\sqrt{y} \pmod{p}$, como mostrado acima. Existe um algoritmo aleatório também de tempo polinomial para calcular as raízes quadradas para primos da forma $4k+1$ [12]. Usando esse algoritmo, é possível selecionar qualquer primo ímpar para p e q.

5.2 SEGURANÇA DO CRIPTOSSISTEMA DE RABIN

É possível demonstrar que esse criptossistema é tão seguro quanto é difícil o problema da fatoração de números inteiros.

Imagine que exista um algoritmo que, em tempo polinomial, retorne uma das raízes de uma mensagem criptografada y qualquer apenas com a chave pública n. Note que $n = pq$, mas os números primos que compõem a chave são desconhecidos. Escolhemos então uma mensagem m qualquer e calculamos $y = m^2 \pmod{n}$. Inserimos no algoritmo acima y e n e então o algoritmo retorna x que é um dos quatro valores possíveis abaixo:

$$\begin{cases} x_1 = m \pmod{n} \\ x_2 = -m \pmod{n} \\ x_3 = \omega m \pmod{n} \\ x_4 = -\omega m \pmod{n} \end{cases}$$

onde $\omega^2 = 1 \pmod{n}$ e $\omega \neq \pm 1 \pmod{n}$.

Teremos com probabilidade $\frac{1}{2}$, uma raiz $x \neq \pm m \pmod{n}$ na qual

$$\begin{aligned} m^2 &= x^2 \pmod{n} \\ m^2 - x^2 &= 0 \pmod{n} \\ (m+x)(m-x) &= 0 \pmod{pq} \end{aligned}$$

Como $m + x \neq 0 \pmod{n}$ e $m - x \neq 0 \pmod{n}$, temos que $m + x = 0 \pmod{p}$ e $m - x = 0 \pmod{q}$ (ou vice versa), pois se isso não ocorresse, teríamos como opção que $m + x = 0 \pmod{n}$ ou $m - x = 0 \pmod{n}$, hipóteses já descartadas de princípio.

Assim, $\text{mdc}(m + x, n) = p$ e $\text{mdc}(m - x, n) = q$.

Podemos concluir que, se existisse um algoritmo para descriptografar o criptossistema de Rabin, poderíamos fatorar $n = pq$ com probabilidade $\frac{1}{2}$ e em tempo polinomial, o que contradiz a conjectura proposta em 2.2.2.

5.3 ENCRIPTAÇÃO HOMOMÓRFICA

Homomorfismo é um termo comumente usado na Matemática para indicar algum tipo de função que preserve a estrutura algébrica, ou seja, podemos operar com os elementos antes ou depois de aplicar o homomorfismo pois a estrutura se mantém. A definição abaixo ajudará a explicar esse conceito.

Definição 5.1 (Encriptação Homomórfica)**.** Um esquema de encriptação $E(k, m)$, onde k é uma chave e m a mensagem, é homomórfico se $E(k, x) \cdot E(k, y) = E(k, x * y)$, onde $\cdot$ e $*$ são operações quaisquer.

Usando uma encriptação homomórfica, é possível para alguém operar com os dados criptografados e fornecer um resultado sem ter conhecimento dos dados. Abaixo veremos um exemplo de uso usando sistemas de votação, mas existem outras possibilidades.

Teorema 5.2. *O criptossistema de Rabin é homomórfico.*

Demonstração. Seja n a chave pública do criptossistema de Rabin descrito anteriormente e sejam a e b duas mensagens quaisquer. Então,

$$\begin{aligned} E(a) &= a^2 \pmod{n} \\ E(b) &= b^2 \pmod{n} \\ E(a) \cdot E(b) &= a^2 \cdot b^2 \pmod{n} \\ E(a) \cdot E(b) &= (a \cdot b)^2 \pmod{n} \\ \therefore E(a) \cdot E(b) &= E(a \cdot b) \quad \square \end{aligned} \tag{5.1}$$

5.3.1 *Sistemas de Votação*

Para utilizar o criptossistema de Rabin na votação, vamos modificá-lo um pouco. Para simplificar, vamos considerar que cada voto v_i só poderá adquirir dois valores: $v_i = 1$, ou seja, "sim"e $v_i = 0$, "não".

Não conhecemos modo de criar um sistema de votação homomórfico usando apenas resíduos quadráticos, então usaremos também outra função conjecturada de mão única que é o logaritmo discreto.

Como no Rabin, precisaremos de dois números primos secretos p e q para obtermos $n = p \cdot q$ e de g, uma raiz primitiva de p. Além disso, precisaremos também de uma chave secreta k. As restrições usuais para p e q se aplicam, mas além disso precisamos que $t < p - 1$, onde t é o total de votantes. O motivo para isso é que g é uma raiz primitiva de p, logo $\{g, g^2, \cdots, g^{p-1}\}$ forma um sistema reduzido de resíduos módulo p, com $p - 1$ elementos diferentes. Logicamente, pode ser que $g^t \pmod{n}$ tenha um período maior que $g^t \pmod{p}$, mas usaremos o menor número como segurança.

Para a encriptação, deve ser escolhido um número aleatório r_i, co-primo com p e q. Para encriptar cada voto v_i, fazemos então

$$E(v_i) = (r_i^2, g^{v_i+k} \cdot r_i), 1 \leq i \leq t$$

A contagem de votos é feita ao multiplicar todos $E(v_i)$ termo a termo, ou seja

$$E(v) = (r_1^2 \cdot r_2^2 \cdot \cdots \cdot r_t^2, g^{v_1+k} \cdot r_1 \cdot g^{v_2+k} \cdot r_2 \cdot \cdots \cdot g^{v_t+k} \cdot r_t) = (r^2, g^{v+t\cdot k} \cdot r) \pmod{n}$$

Ao decifrar $E(v)$ e obter v, que é o total de votos sim, precisamos calcular r com o método descrito em 5.1, e achar $r^{-1} \pmod{n}$ e g^{-kt} para finalmente descobrir $g^{v+kt} \cdot r \cdot r^{-1} \cdot g^{-kt} = g^v \pmod{n}$ ·

Com o intuito de saber o valor de v, devemos utilizar uma tabela para comparar g^v com os valores possíveis. Por esse motivo, v, e consequentemente t, não pode ser um número muito grande pois encontrar v de outra maneira relaciona-se ao problema já discutido em 2.2.1, o problema do logaritmo discreto, que conjectura-se ser uma função de mão única e portanto de difícil solução.

Como exemplo prático, consideremos os números primos $p = 7$ e $q = 11$, com $g = 5$ sendo uma raiz primitiva de p e $k = 4$. Nessa votação, teremos $t = 5$ eleitores. A tabela abaixo sumariza os votos v_i de cada um e r_i, um número aleatório. Após a votação, serão conhecidos apenas os pares ordenados $E(v_i)$.

i	v_i	r_i	$E(v_i) = (r_i^2, g^{v_i} \cdot r_i)$
1	1	36	(64,3)
2	1	75	(4,64)
3	0	51	(60,74)
4	0	47	(53,38)
5	1	26	(60,15)

Para contar os votos, fazemos

$$E(v) = (64 \cdot 4 \cdot 60 \cdot 53 \cdot 60, 3 \cdot 64 \cdot 74 \cdot 38 \cdot 15) \pmod{77} = (4, 8)$$

A partir de $r^2 = 4$, obtemos como raízes $r_1 = 9, r_2 = 2, r_3 = 68, r_4 = 75$, cujos inversos módulo 77 são $r_1^{-1} = 60, r_2^{-1} = 39, r_3^{-1} = 17, r_4^{-1} = 38$ e como conhecemos a chave secreta k e o total de votos t, fazemos $g^{kt} \pmod{77} = 67$, cujo inverso é $g^{-kt} \pmod{77} = 23$.

Multiplicando $x = g^{v+kt} \cdot r \cdot r^{-1} \cdot g^{-kt} \pmod{77}$, obtemos quatro valores, $x_1 = 29, x_2 = 15, x_3 = 48, x_4 = 62$. Comparando com a tabela dos valores de $5^v \pmod{77}$ abaixo, temos que o único valor existente na tabela é 48 que fornece $v = 3$ votos sim. Há métodos para distinguir qual raiz usar, mas nesse caso é bem claro qual o único valor possível.

v	$5^v \pmod{77}$
0	1
1	5
2	25
3	48
4	9
5	45

Alguém que queira descobrir algum voto $E(v_i)$ qualquer sem conhecer os segredos p, q, k deverá descobrir o valor de r a partir de r^2 e o valor de g^k, dois problemas de difícil solução.

6

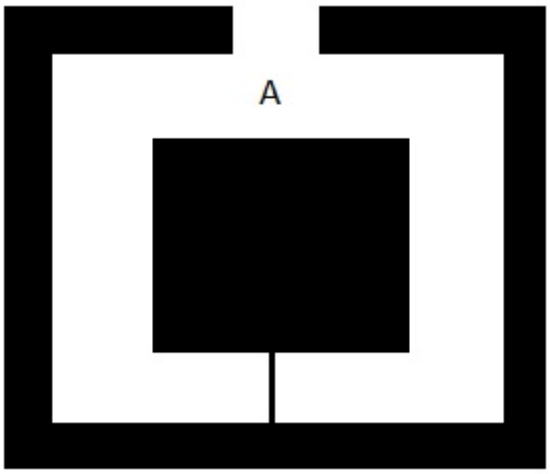
PROVA DE CONHECIMENTO ZERO

Imagine que, para acessar um certo lugar, seja necessário conhecer uma palavra secreta. Ao entrar no lugar, o porteiro pergunta a você qual é a palavra secreta, mas se você falar a palavra secreta todos ao seu redor, inclusive o porteiro, conhecerão o segredo para entrar no lugar. O ideal seria um método para que você mostre que tem o segredo necessário para entrar no lugar, mas sem realmente contar o segredo para ninguém. Uma prova de conhecimento zero é essencialmente isso.

Essa situação é melhor ilustrada por um exemplo criado por Quisquater, Guillou e Berson [9] e que será adaptado a seguir.

Imagine que existe uma caverna conforme a figura a seguir com uma porta na extremidade oposta a entrada que só pode ser aberta com uma palavra secreta. Peggy quer provar para Victor, o verificador, que conhece a palavra secreta, mas sem contar a ele qual é esse segredo. Para tal, eles usam o seguinte mecanismo. Peggy entra na caverna e escolhe ir para um lado, aguardando em frente a porta. Victor então entra na caverna, sem ter visto para que lado Peggy foi, e escolhe entre esquerdo e direito para que Peggy saia por esse lado. Se Victor escolher o lado oposto àquele que Peggy entrou, ela usa a palavra secreta, atravessando a porta e saindo pelo lado escolhido, caso contrário ela apenas volta pelo mesmo caminho. De qualquer maneira, Peggy sempre é capaz de sair da caverna pelo lado desejado por Victor.

Victor pode pensar que é uma grande coincidência Peggy ter saído pelo lado escolhido, pois ela pode ter entrado justo pelo lado que Victor escolheu e não precisou passar pela porta. Isso ocorreria com uma probabilidade de $\frac{1}{2}$. Para se convencer que Peggy possui o segredo, ele pede para que o experimento seja repetido n vezes. Como Peggy conhece o segredo, ela sempre consegue voltar pelo caminho escolhido e a probabilidade que ela o tenha feito por pura sorte é de $\frac{1}{2^n}$ e Victor se convence que ela



conhece a palavra secreta pois percebe a probabilidade de que Peggy tenha passado no teste por sorte é desprezível.

Suponha que Eva estava observando Victor de maneira secreta. Eva não descobre o segredo de Peggy e se for tentar provar para Victor que conhece o segredo ela só conseguirá se, por coincidência, Victor escolher o lado que ela entrou na caverna, o que só acontecerá em $\frac{1}{2}$ das vezes. Após repetir o teste n vezes, espera-se que Eva não seja capaz de voltar pelo lado escolhido em $\frac{n}{2}$ vezes, o que não convencerá Victor que ela possui o segredo.

Existem diversas maneiras de executar esse procedimento matematicamente [13], a seguir será descrita uma usando resíduos quadráticos.

6.1 PROVA DE CONHECIMENTO ZERO COM RAÍZES QUADRADAS MÓDULO N

Como já foi visto antes em 5.2, calcular a raiz quadrada módulo N, quando $N = p \cdot q$ com p e q primos, é tão difícil como fatorar N. Seja y algum quadrado módulo N, co-primo com N, e Peggy quer provar para Victor que ela conhece o segredo que é a raiz quadrada de y módulo N, ou seja, que ela conhece s tal que $s^2 = y \pmod N$, mas ela não quer que Victor descubra s.

Peggy então escolhe um número aleatório r_1 e encontra r_2 tal que $r_1 \cdot r_2 = s \pmod N$, ou seja, $r_2 = r_1^{-1} \cdot s \pmod N$. Em seguida, calcula $x_1 = r_1^2 \pmod N$ e $x_2 = r_2^2 \pmod N$ e envia a Victor x_1 e x_2.

Victor verifica que $x_1 \cdot x_2 = y \pmod N$ e escolhe x_1 ou x_2 para que Peggy forneça sua raiz e verifica que a raiz fornecida é a correta. O procedimento é repetido até que Victor seja convencido que Peggy conhece a raiz s.

Se Peggy conhecer a raiz, ela sempre é capaz de fornecer r_1 e r_2, tomando o cuidado de não repetir os valores de r_1 pois se ela fornecer a Victor duas vezes o mesmo valor x_1 e x_2, ele pode pedir a Peggy as duas raízes e descobrir $s = r_1 \cdot r_2 \pmod N$, justamente o que Peggy está evitando.

Supondo que Peggy não conhece a raiz de y, ela ainda é capaz de enviar a Victor x_1 e x_2 tal que $x_1 \cdot x_2 = y \pmod N$. Ela pode tentar prever qual das duas raízes Victor pedirá, por exemplo x_2 e calcular $x_2 = r_2^2 \pmod N$ e calcular x_1 para que $x_1 \cdot x_2 = y \pmod N$. Ela não é capaz de calcular r_1, pois se fosse poderia calcular s e nesse caso ela é capaz de enganar Victor em $\frac{1}{2}$ das vezes. Após n tentativas, a probabilidade de Victor ser convencido se Peggy não conhece s é desprezível.

Caso tenha alguém observando, por exemplo Eva, ela irá apenas conhecer algumas raízes quadradas aleatórias. Ela será capaz de enganar Victor se ele perguntar a mesma sequência de números x_1 e x_2. Se a ordem for mudada, Eva não é capaz de usar a informação obtida para calcular novas raízes. Essa é a característica importante de uma prova de conhecimento zero, já que o verificador ou qualquer outro observador não conseguem nenhum conhecimento adicional a partir da resposta de Peggy.

Obviamente, todos tem o conhecimento de N mas não dos primos p e q que o originaram. Caso contrário, é relativamente simples calcular as raízes quadradas de qualquer y, como visto anteriormente em 5.1.

6.2 PROTOCOLO FEIGE-FIAT-SHAMIR

Esse protocolo foi desenvolvido por Uriel Feige, Amos Fiat e Adi Shamir em 1988 [7] e usa uma ideia similar ao capítulo anterior, diminuindo o número de comunicações necessárias. Para isso precisamos novamente de $N = p \cdot q$, o produto de dois números primos desconhecidos.

Peggy tem como segredo os números $s_1, \ldots, s_k$, todos co-primos com N. Victor recebe os números $v_i = s_i^{-2} \pmod N, 1 \leq i \leq k$, os inversos módulo N de $s^2 \pmod N$ e deve verificar se Peggy realmente possui o segredo. Para tal, são executados os seguintes passos:

1. Peggy escolhe um inteiro aleatório r e envia a Victor $x = r^2 \pmod{N}$.
2. Victor escolhe e manda para Peggy uma cadeia de números $b_1, \ldots, b_k$, onde $b_i \in \{0, 1\}$.
3. Peggy devolve a Victor o número y tal que $y = r \cdot s_1^{b_1} \cdot s_2^{b_2} \ldots s_k^{b_k} \pmod{N}$.
4. Victor verifica que $x = y^2 \cdot v_1^{b_1} \cdot v_2^{b_2} \ldots v_k^{b_k} \pmod{N}$

Repete-se a sequência quantas vezes forem necessárias, sempre escolhendo-se um r diferente.

Percebe-se que, para $k = 1$, Peggy deve informar r ou $r \cdot s_1$, dois números cujo quociente é uma raiz quadrada módulo N de v_1, a mesma ideia do processo anterior mas com quocientes ao invés de produto.

Para k maior, é fácil para Peggy calcular y se ela possui o segredo e o único conhecimento que Victor consegue é uma raiz de um número da forma $x = y^2 \cdot v_1^{b_1} \cdot v_2^{b_2} \ldots v_k^{b_k} \pmod{N}$, o que não é suficiente para descobrir o segredo $s_1, \ldots, s_k$.

Se Eva, que não possui o segredo, quiser se passar por Peggy, precisará adivinhar qual a sequência de números $b_1, \ldots, b_k$ Victor pedirá antes de enviar x, pois como não sabe calcular o inverso módulo N das raízes quadradas de $v_1, \ldots, v_k$, ela precisará escolher $x = y^2 \cdot v_1^{b_1} \cdot v_2^{b_2} \ldots v_k^{b_k}$ e retornar a Victor o valor y, que passará no teste.

Como há 2^k possíveis valores de b_i, ao repetir esse esquema t vezes, a probabilidade que Eva engane Victor será de $\frac{1}{2^{kt}}$, que é desprezível.

Numericamente, vamos pegar como exemplo $N = 16637$. Os primos 127 e 131 são os fatores de N mas são secretos.

Peggy tem como segredo 5 números $S = (3994, 2670, 3419, 8487, 2944)$ e Victor recebe como verificador os números $V = (3290, 7479, 10842, 15785, 9138)$. Podemos verficar que $v_i \cdot s_i^2 = 1 \pmod{1}6637$ com $1 \leq i \leq 5$.

Victor então pede a Peggy para verificar se ela possui a senha. Os seguintes passos são executados:

1. Peggy escolhe um número aleatoriamente, $r = 1042$, e informa a Victor o valor $x = 4359$ que é o resultado de $r^2 \pmod{16637}$.
2. Victor escolhe uma cadeia de 5 bits b_i e envia a Peggy. Nesse exemplo, $b_i = (0, 1, 1, 1, 0)$.

3. Peggy calcula $y = 1042 \cdot 3994^0 \cdot 2670^1 \cdot 3419^1 \cdot 8487^1 \cdot 2944^0 \pmod{16637}$ e devolve a Victor o resultado $y = 6365$.
4. Victor verifica que de fato $x = 6365^2 \cdot 3290^0 \cdot 7479^1 \cdot 10842^1 \cdot 15785^1 \cdot 9138^0 \pmod{16637} = 4359$.

Se Victor não ficou contente e acha que pode ter sido enganado, uma probabilidade de $\frac{1}{2^5}$ conforme visto anteriormente, ele pode pedir uma nova verificação. Peggy só precisa escolher um novo número r e repetir o processo.

7

APLICAÇÃO EM SALA DE AULA

Como aplicação em sala de aula, decidi trabalhar de forma lúdica um leilão. O objetivo dessa aula é motivar os alunos e mostrar uma aplicação da álgebra abstrata discreta para alunos de 8^o e 9^o anos do ensino fundamental.

O processo descrito abaixo funciona para leilões e licitações. A diferença é que, enquanto no leilão ganha quem der o maior lance, na licitação ganha quem ofertar o menor preço.

Antes do jogo ser feito, é necessário explicar aos alunos alguns conceitos de aritmética modular, mostrar como funciona o resto das divisões de números inteiros e que esse resto é importante em um grande número de situações, visto que a maioria dos alunos tem por hábito efetuar a divisão de números racionais (utilizando a vírgula) mesmo quando não convém ao problema.

Os alunos são divididos em grupos e a cada grupo é dado um mesmo número arbitrário C de créditos. Esses créditos poderão ser usados para comprar pontos por meio de um leilão aberto. Após um número R de rodadas, a equipe que tiver mais pontos ganhará o jogo.

A tática deve ser que cada grupo dê o maior lance possível, mas se gastar todos os seus créditos de início não terá como ganhar os lances posteriores. Cada grupo deverá apresentar seu lance de forma pública, mas se os outros grupos souberem o lance dos concorrentes antes de divulgar o seu, podem alterar seu lance. Para evitar isso, cada grupo irá criptografar seu lance usando o seguinte processo:

1. Cada grupo escolhe seu lance L em segredo.

2. Os grupos calculam $H_i = L_i^2 \pmod N$, sendo i o número de cada grupo, onde N é um número fornecido previamente pelo professor e $N = p \cdot q$, onde p e q são números primos desconhecidos pelos alunos.
3. Cada grupo apresenta H_i para os demais.
4. Após todos conhecerem H_i, todos revelam L_i e podem verificar que $H_i = L_i^2 \pmod N$.

Alguns cuidados devem ser tomados com a escolha dos números. Por exemplo, $N > 2.C$, pois algum grupo poderia dar uma lance L e dizer que seu lance original era $N - L$ pois $L^2 = (N - L)^2 \pmod N$.

Também pode acontecer de um grupo testar vários valores para descobrir qual lance seu oponente mandou. Para evitar isso, considere que o lance máximo L tem M casas decimais. O grupo escolhe um número Y e calcula $H = (Y \cdot 10^M + L)^2 \pmod N$. Esse processo aumenta um pouco os cálculos, mas torna quase impossível descobrir por tentativa e erro o valor L escolhido. Ao apresentar o valor $Y \cdot 10^M + L$ para verificação, os M últimos dígitos corresponderão ao lance L.

Vamos exemplificar usando $N = 437$. A fatoração desse número é $19 \cdot 23$ mas não será fornecida aos alunos.

Serão feitos quatro grupos e cada grupo receberá $C = 100$ créditos para gastar em $R = 4$ rodadas. Cada um dos grupos deverá criar uma estratégia para vencer. Após a primeira rodada, os lances pensados por cada grupo e os valores H calculados são:

	Grupo 1	Grupo 2	Grupo 3	Grupo 4
L	25	35	10	40
Y	0	0	3	0
$Y.10^3 + L$	25	35	3010	40
H	188	351	216	289

Note que apenas o grupo 3 viu necessidade de usar o método alternativo pois $10^2 = 100 \pmod{437}$ e seria fácil para os demais descobrirem seu lance. Quando todos revelarem seus lances, o grupo 4 ganhará essa rodada e a segunda rodada começará. Lembrando que após essa rodada, todos os grupos ainda dispõem de 100 créditos para a compra com exceção do grupo 4 que possui apenas 60 créditos.

É interessante mencionar aos alunos que calcular as raízes módulo N, sendo N um número composto, é difícil se seus fatores forem grandes, mas que há métodos de calcular as raízes módulo p, sendo p um número primo.

Dependendo do andamento da atividade, pode ser interessante que o professor mostre, após o leilão, que ele (e só ele, que conhece os fatores p e q de N) consegue obter as raízes, sendo necessária uma autoridade confiável (no caso, o professor) para escolher p e q de forma a fornecer um N cuja fatoração seja desconhecida por todos.

BIBLIOGRAFIA

[1] BARKAN, E.; BIHAM, E.; KELLER, N. Instant ciphertext-only cryptanalysis of GSM encrypted communication. In: ANNUAL INTERNATIONAL CRYPTOLOGY CONFERENCE, 23., 2003, Santa Barbara. **Proceedings...** Berlin: Springer, 2003. p. 600-616.

[2] BLUM, L.; BLUM M.; SHUB M. A simple unpredictable pseudo-random number generator. **SIAM Journal on Computing**, v. 15, n. 2, p. 364-383, 1986.

[3] BLUM, M.; MICALI, S. How to generate cryptographically strong sequences of pseudorandom bits. **SIAM Journal on Computing**, v. 13, n. 4, p. 850-864, 1984.

[4] CHAUM, D.; VAN HEIJST, E.; PFITZMANN, B. Cryptographically strong undeniable signatures, unconditionally secure for the signer. **Advances in Cryptology - CRYPTO 91**, p. 470-484, 1992.

[5] DELFS, H.; KNEBL, H. **Introduction to Cryptography**: Principles ans Applications. 2. ed. Berlin Heidelberg: Springer, 2007. 367 p.

[6] FALEIROS, A. C. **Criptografia.** São Carlos: Sociedade Brasileira de Matemática Aplicada e Computacional, 2011. 138 p.

[7] FEIGE, U.; FIAT, A.; SHAMIR, A. Zero-knowledge proofs of identity. **Journal of Cryptology**, v. 1, n. 2, p. 77-94, 1988.

[8] KATZ, J.; LINDELL, Y. **Introduction to Modern Cryptography.** 1 ed. CRC Press, 2007. 498 p.

[9] QUISQUATER, J. J.; GUILLOU, L. C.; BERSON, T. A. How to explain zero-knowledge protocols to your children. **Advances in Cryptology - CRYPTO 89**, p. 628-631, 1990.

[10] RABIN, M. O. **Digitalized signatures and public-key functions as intractable as factorization.** MIT Laboratory for Computer Science, 1979. 16 p.

[11] SHANNON, C. E. Communication theory of secrecy systems. **Bell System Technical Journal**, v. 28, n. 4, p. 656-715, 1949.

[12] STINSON, D. R. **Cryptography**: Theory and Practice. 3 ed. Chapman Hall/CRC, 2006. 593 p.

[13] TRAPPE, W; WASHINGTON, L. C. **Introduction to Cryptography with Coding Theory.** 2. ed. Pearson Education International, 2006. 577 p.

[14] VON NEUMANN, J. Various techniques used in connection with random digits. **National Bureau of Standards Applied Mathematics Series**, n. 12, p. 36-38. 1961.