GBC083 – Segurança da Informação
Aula 1 - Introdução à Criptografia

Prof. Marcelo Keese Albertini

Segurança da Informação - Metas

- Confidencialidade
  - Criptografia clássica (até 1970)
- Integridade
  - Criptografia moderna = Confidencialidade + Integridade
- Disponibilidade
  - Prevenção contra ataques de negação de serviços
  - Prevenção contra perda de dados

Criptografia moderna

- Comunicação segura (SSH, HTTPS)
- Armazenamento seguro (TrueCrypt)
- Assinatura digital (RSA)
- Comunicação anônima (mix net, tor)
- Dinheiro digital anônimo (bitcoin)
- Contratos inteligentes (ethereum)
- Protocolos de eleição (Neff e Chaum) e leilões privados
- Computação segura entre múltiplos atores (homomorfismos)

Criptografia clássica

- Até os anos 1970
  - usa informação secreta (uma chave) compartilhada entre somente as partes comunicantes
- Criptografia de chave privada
  - também chamada de chave secreta, chave compartilhada ou chave simétrica

Criptografia de chave privada

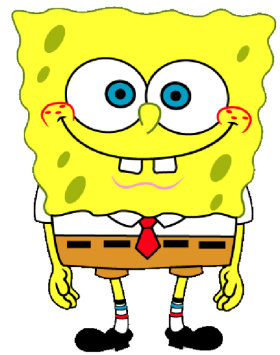
Texto em claro m = "oi bob como vai"

chave k

Encriptação:
$c \leftarrow Enc_k(m)$

Texto cifrado:
c = "alkshdaioshfahr"

chave k



Decriptação:
$m \leftarrow Dec_k(c)$

Criptografia de chave privada

k

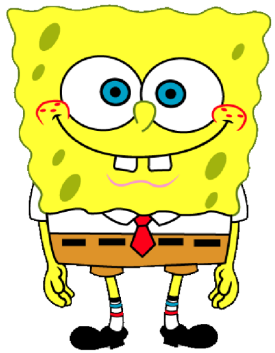
m

$c \leftarrow Enc_k(m)$

c

c

k



$m \leftarrow Dec_k(c)$

Criptografia de chave privada

- Um esquema de criptografia de chave-privada é definido por uma espaço de mensagens $\mathbf{M}$ e algoritmos (Gen, Enc, Dec):
  - Gen (algoritmo de geração de chave): cria k
  - Enc (algoritmo de encriptação): recebe chave k e mensagem $m \in \mathbf{M}$ como entrada; produz texto cifrado c

$$c \leftarrow Enc_k(m)$$

  - Dec (algoritmo de decriptação): recebe chave k e texto cifrado c como entrada e produz m ou "erro"
- Para todo $m \in \mathbf{M}$ e k produzido por Gen, $Dec_k(Enc_k(m)) = m$

A cifra de deslocamento

- Considere encriptar texto em inglês
- Associar ‘a’ com 0. ‘b’ com 1; . . .; ‘z’ com 25
- $k \in \{0, \ldots, 25\}$
- Para encriptar usando chave k, deslocar toda letra do texto limpo em k posições “dando a volta” ao chegar em ‘z’
- Para decriptar é só fazer o reverso
  - helloworld $\otimes$ ccccccccccc $=$ jgnnqyqtnfb

Aritmética modular

- $x = x' \mod N$ se e só se N divide $x - x'$
- $[x \mod N]$ = o resto quando x é dividido por N
  - Isto é, o único valor $x' \in \{0, \ldots, N-1\}$ tal que $x = x' \mod N$
- $25 = 35 \mod 10$
- $25 \neq [35 \mod 10]$
- $5 = [35 \mod 10]$
- Cuidado! No C e no Java, o operador de resto % retorna números negativos!

A cifra de deslocamento, formalmente

- $\mathbf{M} = \{$ string sobre o alfabeto minúsculo inglês $\}$
- Gen: escolher $k \in \{0, \ldots, 25\}$
- $Enc_k(m_1 \ldots m_t)$: obter $c_1 \ldots c_t$ onde $c_i \leftarrow [m_i + k \mod 26]$
- $Dec_k(c_1 \ldots c_t)$: obter $m_1 \ldots m_t$ onde $m_i \leftarrow [c_i - k \mod 26]$

A cifra de deslocamento é segura?

- Não – existem apenas 26 chaves possíveis
  - Dado um texto cifrado, tente decriptar com todo chave possível
  - Se texto cifrado é longo (e texto original é em inglês comum), somente um texto decriptado fará sentido

Exemplo

- Texto cifrado: uryybjbeyq
- Tentar cada chave:
  - tqxxauadxp
  - spwwzhzcwo
  - ...
  - helloworld

Princípio de Kerckhoffs

- O esquema de encriptação não deve ser secreto
  - O único segredo é a chave
  - A chave deve ser escolhida aleatoriamente e mantida em sigilo
- Alguns argumentos em favor desse princípio
  - Mais fácil manter chave secreta que algoritmo secreto
    - Algoritmo quebrado leva a todos os segredos serem expostos
  - Mais fácil mudar chave que mudar algoritmo
    - Chave descuberta leva a alguns segredos expostos
- Padronização
  - Facilita instalação do esquema
  - Validação pública

Princípio do espaço suficiente de chaves

- Princípio de segurança informal
- Espaço de chave deve ser grande o suficiente para prevenir "força-bruta" em ataques de busca exaustiva

Cifra de Vigenère

- A chave da Cifra de Vigenère é uma string e não um caractere como antes
- Para encriptar, *Enc*, deslocar cada caractere no texto limpo pela quantidade indicada pelo próximo caractere da chave
  - Voltar ao início da chave se necessário
- Para desencriptar, *Dec*, reverter processo
- tellhimaboutme $\oplus$ cafecafecafeca = veqpjiredozxoe

A cifra de Vigenère

- Tamanho do espaço de chaves
  - Se chave é de 10 carateres então tamanho do espaço de chaves é 26^{10}
  - Se chaves é strings de N caracteres então espaço da chaves tem tamanho 26^N
  - Busca por força-bruta é cara e por isso impossível
- A cifra de Vigenère é segura?

Implementação - ASCII

Hex	nibble	Decimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7

Hex	nibble	Decimal
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

- nibble = pedaço de 4 bits
- 1 nibble = 1 dígito hexadecimal

Hexadecimal – base 16

- $0x10$
  - $0x10 = 16 \times 1 + 0 = 16$
  - $0x10 = 0001\,0000$
- $0xAF$
  - $0xAF = 16 \times A + F = 16 \times 10 + 15 = 175$
  - $0xAF = 1010\,1111$

ASCII

- Caracteres representados em ACII
  - 1 byte por caractere = 2 dígitos hexas por caractere

Hex	Dec	Char		Hex	Dec	Char	Hex	Dec	Char	Hex	Dec	Char
0x00	0	NULL	null	0x20	32	Space	0x40	64	@	0x60	96	`
0x01	1	SOH	Start of heading	0x21	33	!	0x41	65	A	0x61	97	a
0x02	2	STX	Start of text	0x22	34	"	0x42	66	B	0x62	98	b
0x03	3	ETX	End of text	0x23	35	#	0x43	67	C	0x63	99	c
0x04	4	EOT	End of transmission	0x24	36	$	0x44	68	D	0x64	100	d
0x05	5	ENQ	Enquiry	0x25	37	%	0x45	69	E	0x65	101	e
0x06	6	ACK	Acknowledge	0x26	38	&	0x46	70	F	0x66	102	f
0x07	7	BELL	Bell	0x27	39	'	0x47	71	G	0x67	103	g
0x08	8	BS	Backspace	0x28	40	(	0x48	72	H	0x68	104	h
0x09	9	TAB	Horizontal tab	0x29	41	)	0x49	73	I	0x69	105	i
0x0A	10	LF	New line	0x2A	42	*	0x4A	74	J	0x6A	106	j
0x0B	11	VT	Vertical tab	0x2B	43	+	0x4B	75	K	0x6B	107	k
0x0C	12	FF	Form Feed	0x2C	44	,	0x4C	76	L	0x6C	108	l
0x0D	13	CR	Carriage return	0x2D	45	-	0x4D	77	M	0x6D	109	m
0x0E	14	SO	Shift out	0x2E	46	.	0x4E	78	N	0x6E	110	n
0x0F	15	SI	Shift in	0x2F	47	/	0x4F	79	O	0x6F	111	o
0x10	16	DLE	Data link escape	0x30	48	0	0x50	80	P	0x70	112	p
0x11	17	DC1	Device control 1	0x31	49	1	0x51	81	Q	0x71	113	q
0x12	18	DC2	Device control 2	0x32	50	2	0x52	82	R	0x72	114	r
0x13	19	DC3	Device control 3	0x33	51	3	0x53	83	S	0x73	115	s
0x14	20	DC4	Device control 4	0x34	52	4	0x54	84	T	0x74	116	t
0x15	21	NAK	Negative ack	0x35	53	5	0x55	85	U	0x75	117	u
0x16	22	SYN	Synchronous idle	0x36	54	6	0x56	86	V	0x76	118	v
0x17	23	ETB	End transmission block	0x37	55	7	0x57	87	W	0x77	119	w
0x18	24	CAN	Cancel	0x38	56	8	0x58	88	X	0x78	120	x
0x19	25	EM	End of medium	0x39	57	9	0x59	89	Y	0x79	121	y
0x1A	26	SUB	Substitute	0x3A	58	:	0x5A	90	Z	0x7A	122	z
0x1B	27	FSC	Escape	0x3B	59	;	0x5B	91	[	0x7B	123	{
0x1C	28	FS	File separator	0x3C	60	<	0x5C	92	\	0x7C	124	|
0x1D	29	GS	Group separator	0x3D	61	=	0x5D	93	]	0x7D	125	}
0x1E	30	RS	Record separator	0x3E	62	>	0x5E	94	^	0x7E	126	~
0x1F	31	US	Unit separator	0x3F	63	?	0x5F	95	_	0x7F	127	DEL

ASCII

- ‘1’ $= 0x31 = 00110001$
- ‘F’ $= 0x46 = 01000110$

Base64

- Base64 é codificação para representar binário em ASCII.
- Texto: M a n
- ASCII: 77 (0x4d), 97 (0x61), 110 (0x6e)
- Padrão de 8 bits : 01001101 01100001 01101110
- Padrão de 6 bits : 010011 010110 000101 101110
  - $2^6 = 64$
- Número de índice: 19 22 5 46
  - A=0 . . . Z=25, a=26. . . z=51, 0=52 . . . 9=61, + = 62, / = 63
- Base64: T W F u
  - Caractere especial '=' indica padding de 1 byte para quando texto não tiver número de bits múltiplo de 6
  - Possível usar 1 byte '=' ou 2 '==' de padding

Representação de dados

- Como armazenar o valor $0x1F$ em um arquivo?
- Uma opção
  - hexadecimal puro: armazenar 00011111
    mas para visualizar não é possível usar um editor de textos
- Outra opção
  - texto: guardar caracteres ASCII $1F$ ou em Base64
    - Ou seja, em binário: 001100011000110
    - Se abrimos com editor de texto, veremos $1F$
    - Ao ler de volta para descriptografar, necessário converter de volta para $0x1F$

Exercícios

- Faça um programa para converter uma string em hexa para base64.
- Usar somente hexa/base64 para produzir arquivo texto fácil de ler
- Em códigos sempre opere em binário (`byte` em Java, e `uint8_t` em C)
- Implemente a cifra de Vigenère usado a entrada e a saída de texto cifrado em hexa
  - Encriptação e decriptação
- Implemente um ataque de força bruta contra a cifra de Vigenère