nedeljainformatike v2.0

Kriptografija

Andrej Ivašković, Aleksa Marušić

Matematička gimnazija
Nedelja informatike v2.0

14. decembar 2015.

Čemu ovo predavanje

Najpopularnije predavanje prethodne Nedelje je bilo *Računarska bezbednost i lozinke*. Sada ćemo ovoj temi pristupiti iz drugog, nešto starijeg ugla.

Sveprisutne ideje, posebno u području E-Commerce.

Tema je aktuelna: vlade nekih država nameravaju da "regulišu korišćenje kriptografije".

Plan predavanja

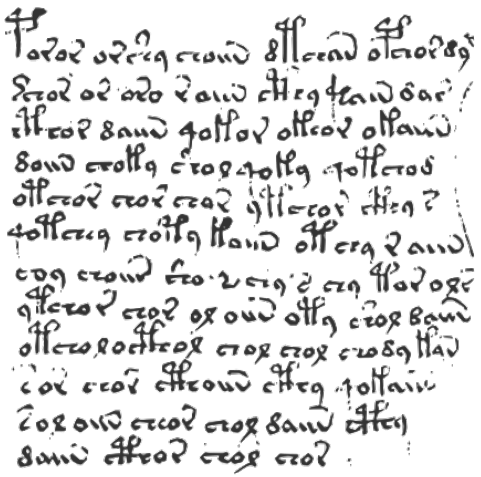
1. Osnovni pojmovi kriptografije
2. Istorijski pregled šifara (uz uvođenje neophodnih pojmova iz teorije brojeva)
3. Javni ključ i zajednički ključ

---

4. Savremeni kriptografski algoritmi i protokoli

Pitanja na koja želimo da odgovorimo

- Šta je to **šifra**?
- Kako da otkrijemo da `NDEAANCSPIEXAVRX` baš znači `NAPADNI VECERAS`?
- Šta bi bila **kriptografija**?
- Kakva treba da bude šifra?
- Čemu nam služe šifre?



Šifra i kod

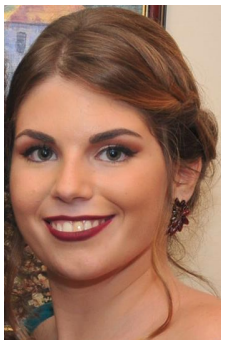
Šifra:

- procesira *znake* (ili "blokove" znakova: **blok šifre**) pomoću utvrđenog algoritma: `"NAPADNI IVASKOVICA VECERAS"` $\rightarrow$ `"XBTFSARETASDHTRDFFVV"`;
- često poznata matematička pozadina;
- nije komplikovano implementirati na računaru.

Kod:

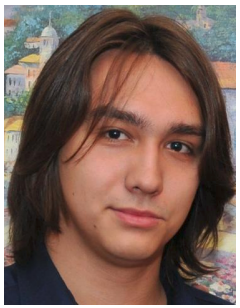
- bavi se *semantikom* (značenjem) poruke: `"NAPADNI IVASKOVICA VECERAS"` $\rightarrow$ `"KUPI RATLUK, KIKIRIKI I PISTACO"`;
- sistem koji nema matematičku osnovu;
- teško ostvarivo kodiranje i dekodiranje pomoću računara.

Akteri

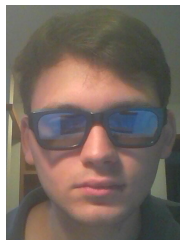


Slika: Alisa, fina devojka

Slika: Eva (*Eve*, eng. *eavesdropper*), radoznala devojka



Slika: Bob, fin mladić



Slika: Melet (*Mallet*, eng. *malicious*), osoba sa lošim namerama

Osnovne definicije

- **Kriptografija:** disciplina koja izučava načine ostvarivanja bezbedne i tajne komunikacije između dve strane (dakle, bez učešća i uticaja treće strane).
- **Šifra:** kriptografski algoritam.
- **Kriptoanaliza:** proučavanje nekog kriptografskog algoritma/sistema radi otkrivanja njegovih aspekata koji bi trebalo da budu tajni.
- U daljoj priči ćemo videti u kojoj meri treba da znamo da čuvamo tajne.

Opšti postupak slanja poruke

- **Plaintext:** poruka koju smo napisali i želimo da je primalac (i samo primalac) vidi.
- **Ciphertext:** odnosno **šifrat**, transformisan plaintext, trebalo bi da bude beznačajan Evi i Meletu.
- **Transport:** smatramo da nije bezbedan i da je prisluškivanje veoma lako.
- **Enkripcija:** pretvaranje plaintext u ciphertext.
- **Dekripcija:** pretvaranje ciphertext u plaintext.

Nekada davno...

Primer transpozicijske šifre

1. Počnimo od `NAPADNIVECERASXX` i upišimo to u tabelu 4×4, š leva na desno".

N	A	P	A
D	N	I	V
E	C	E	R
A	S	X	X

Primer transpozicijske šifre

1. Počnimo od `NAPADNIVECERASXX` i upišimo to u tabelu 4×4, š leva na desno".

N	A	P	A
D	N	I	V
E	C	E	R
A	S	X	X

2. Kada pročitamo "odozgo na dole", dobije se `NDEAANCSPIEXAVRX`.

Primer transpozicijske šifre

1. Počnimo od `NAPADNIVECERASXX` i upišimo to u tabelu 4×4, š leva na desno".

N	A	P	A
D	N	I	V
E	C	E	R
A	S	X	X

2. Kada pročitamo "odozgo na dole", dobije se `NDEAANCSPIEXAVRX`.
3. Dešifrovanje se vrši pomoću iste tabele!

Primer transpozicijske šifre

1. Počnimo od `NAPADNIVECERASXX` i upišimo to u tabelu 4×4, š leva na desno".

N	A	P	A
D	N	I	V
E	C	E	R
A	S	X	X

2. Kada pročitamo "odozgo na dole", dobije se `NDEAANCSPIEXAVRX`.
3. Dešifrovanje se vrši pomoću iste tabele!

- Reč je o šiframa kod kojih je šifrat permutacija plaintexta.

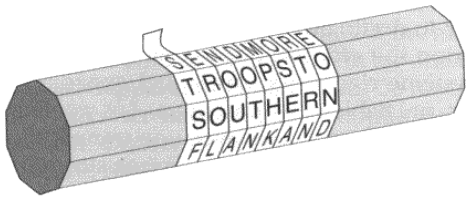
Primer transpozicijske šifre

1. Počnimo od `NAPADNIVECERASXX` i upišimo to u tabelu 4×4, š leva na desno".

N	A	P	A
D	N	I	V
E	C	E	R
A	S	X	X

2. Kada pročitamo "odozgo na dole", dobije se `NDEAANCSPIEXAVRX`.
3. Dešifrovanje se vrši pomoću iste tabele!

- Reč je o šiframa kod kojih je šifrat permutacija plaintexta.
- Slično, spartanska skitala.



Cezarova šifra i ROT13

1. Gaj Julije Cezar se koristio narednom šifrom (pomeranje za 3 "ulevo", pri čemu latinska abeceda tada nije imala U, Y, Z, dok J i W uopšte ne postoje):

znak	A	B	C	D	...	X
zameni sa	T	V	X	A	...	S

Cezarova šifra i ROT13

1. Gaj Julije Cezar se koristio narednom šifrom (pomeranje za 3 "ulevo", pri čemu latinska abeceda tada nije imala U, Y, Z, dok J i W uopšte ne postoje):

znak	A	B	C	D	...	X
zameni sa	T	V	X	A	...	S

2. Tako SALVE $\rightarrow$ PTHRB.

Cezarova šifra i ROT13

1. Gaj Julije Cezar se koristio narednom šifrom (pomeranje za 3 "ulevo", pri čemu latinska abeceda tada nije imala U, Y, Z, dok J i W uopšte ne postoje):

znak	A	B	C	D	...	X
zameni sa	T	V	X	A	...	S

2. Tako SALVE $\rightarrow$ PTHRB.
3. Dešifrovanje se vrši pomerajem za 3 "udesno".

Cezarova šifra i ROT13

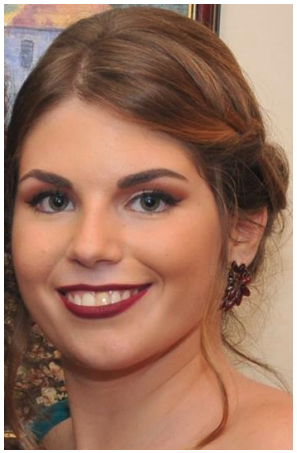
1. Gaj Julije Cezar se koristio narednom šifrom (pomeranje za 3 "ulevo", pri čemu latinska abeceda tada nije imala U, Y, Z, dok J i W uopšte ne postoje):

znak	A	B	C	D	...	X
zameni sa	T	V	X	A	...	S

2. Tako SALVE $\rightarrow$ PTHRB.
3. Dešifrovanje se vrši pomerajem za 3 "udesno".

- Varijanta sa savremenom engleskom abecedom od 26 karaktera: pomeraj za 13. Program pod Unix-om: `rot13`.

Oprez!



- Pretpostavimo da je Eva presrela poruku koju je Alisa poslala Bobu.
- Međutim, ona nema nikakvu predstavu šta je `SFAFJLPBRMBQ`.
- Zamislimo da je otkrila da njih dvoje koriste Cezarovu šifru i da je saznala da će se "videti u pet". Koje su posledice? Kako da se odbranimo od ovoga?

Opasnost!

U još gorem scenariju, Melet bi presreo, a zatim i "hakovao" originalnu poruku i poslao Bobu "lažnu" poruku u kojoj mu se javlja da bi trebalo da dođe u šest!

Slika: for teh evulz

Monoalfabetske supstitucijske šifre

- **Monoalfabetske supstitucije** su određene nekom permutacijom znakova `A, B, C, ..., Z`:

znak	A	B	C	D	...	Z
zameni sa	R	F	M	N	...	K

Monoalfabetske supstitucijske šifre

- **Monoalfabetske supstitucije** su određene nekom permutacijom znakova `A, B, C, ..., Z`:

znak	A	B	C	D	...	Z
zameni sa	R	F	M	N	...	K

- Uz dovoljno velik šifrat vezan za jednu permutaciju, sve ovakve šifre mogu da se razbiju **analizom frekvencija** (Al-Kindi, IX v. n. e.).

Monoalfabetske supstitucijske šifre

- **Monoalfabetske supstitucije** su određene nekom permutacijom znakova `A, B, C, ..., Z`:

znak	A	B	C	D	...	Z
zameni sa	R	F	M	N	...	K

- Uz dovoljno velik šifrat vezan za jednu permutaciju, sve ovakve šifre mogu da se razbiju **analizom frekvencija** (Al-Kindi, IX v. n. e.).
- Učestali monografi u engleskom jeziku (sortirani): `ETAOINSHRDLCU`. U srpskom: `AIOENSTRUMDVK`.

Monoalfabetske supstitucijske šifre

- **Monoalfabetske supstitucije** su određene nekom permutacijom znakova `A, B, C, ..., Z`:

znak	A	B	C	D	...	Z
zameni sa	R	F	M	N	...	K

- Uz dovoljno velik šifrat vezan za jednu permutaciju, sve ovakve šifre mogu da se razbiju **analizom frekvencija** (Al-Kindi, IX v. n. e.).
- Učestali monografi u engleskom jeziku (sortirani): `ETAOINSHRDLCU`. U srpskom: `AIOENSTRUMDVK`.
- Nekada se posmatraju se i česti digrafi: `TH, HE, IN, ER, AN, RE`.

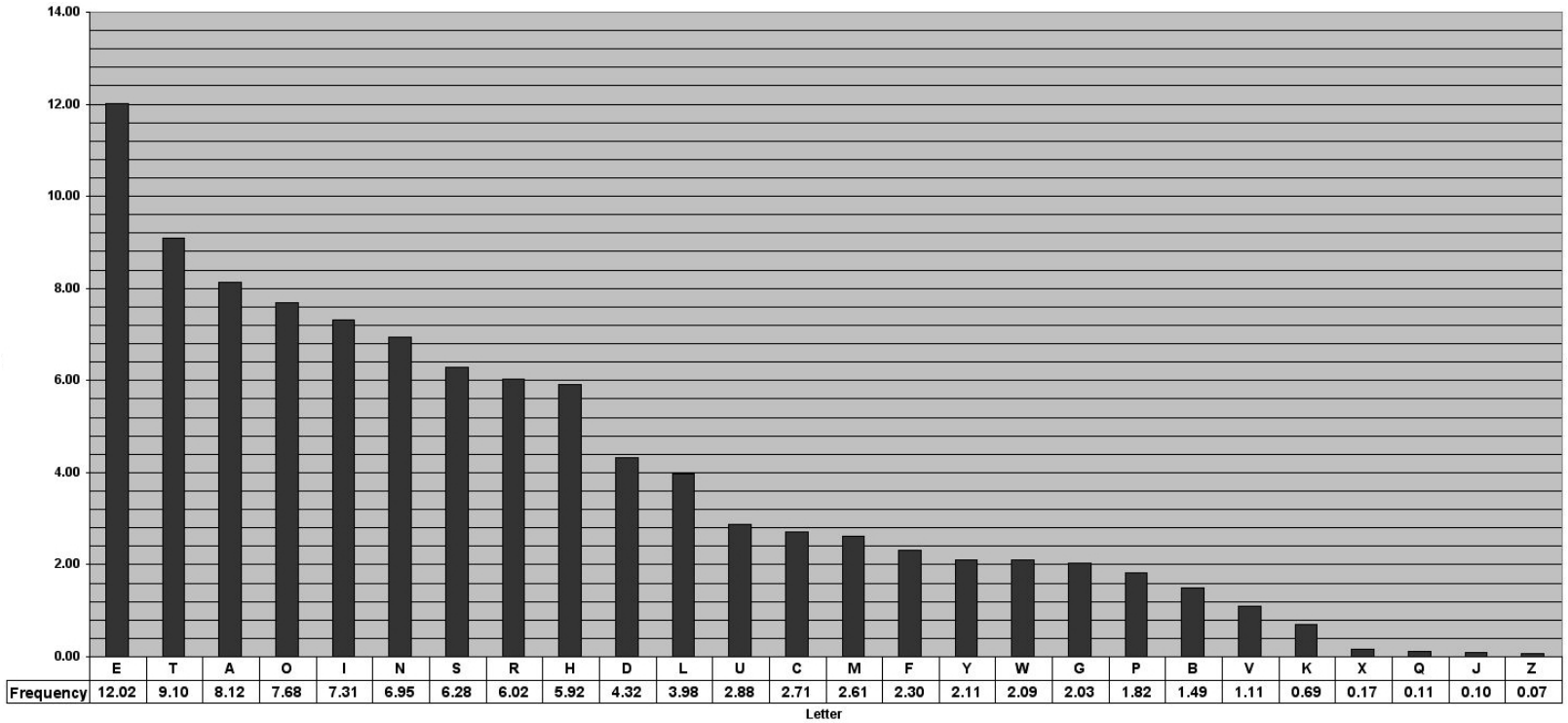
Monoalfabetske supstitucijske šifre

- **Monoalfabetske supstitucije** su određene nekom permutacijom znakova `A, B, C, ..., Z`:

znak	A	B	C	D	...	Z
zameni sa	R	F	M	N	...	K

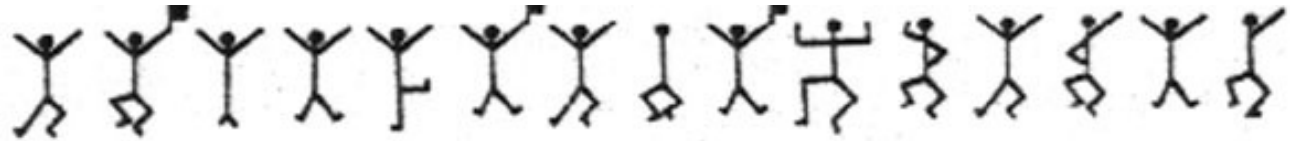
- Uz dovoljno velik šifrat vezan za jednu permutaciju, sve ovakve šifre mogu da se razbiju **analizom frekvencija** (Al-Kindi, IX v. n. e.).
- Učestali monografi u engleskom jeziku (sortirani): `ETAOINSHRDLCU`. U srpskom: `AIOENSTRUMDVK`.
- Nekada se posmatraju se i česti digrafi: `TH, HE, IN, ER, AN, RE`.
- Protiv ovoga se borimo korišćenjem homonimnih supstitucija: `E` se pishe i kao `G` i kao `L`, često se uvode i cifre, `W` se predstavlja kao `VV`...

Frekvencije slova u engleskom jeziku



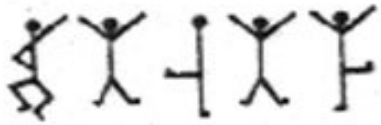
	E	T	A	O	I	N	S	R	H	D	L	U	C	M	F	Y	W	G	P	B	V	K	X	Q	J	Z
Frequency	12.02	9.10	8.12	7.68	7.31	6.95	6.28	6.02	5.92	4.32	3.98	2.88	2.71	2.61	2.30	2.11	2.09	2.03	1.82	1.49	1.11	0.69	0.17	0.11	0.10	0.07

The Adventure of the Dancing Men (Sherlock Holmes)

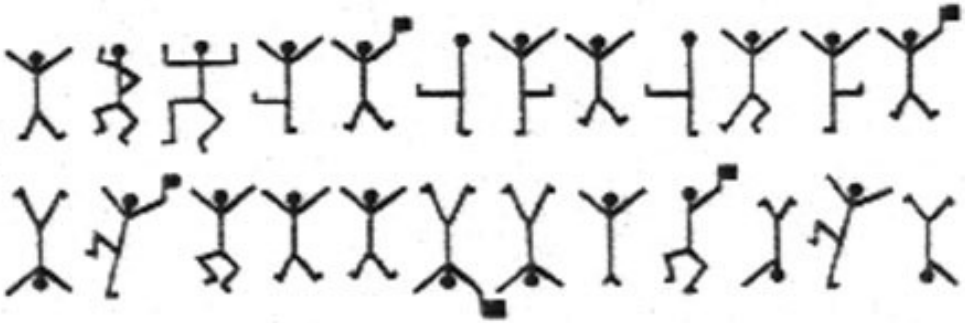


criminal's message (1)

criminal's message (2)



Elsie's reply



criminal's message (3)

Vižnerova šifra

1. Pored poruke, potrebna nam je i **ključna reč**. Ona se zatim produži na sledeći način i dobijemo **ključ**:

ključna reč	KROV
ključ	KROVKROVKR
poruka	DOLAZISADA

Vižnerova šifra

1. Pored poruke, potrebna nam je i **ključna reč**. Ona se zatim produži na sledeći način i dobijemo **ključ**:

ključna reč	KROV
ključ	KROVKROVKR
poruka	DOLAZISADA

2. Pri obradi svakog znaka u poruci koristimo posebnu monoalfabetsku supstituciju koju određuje odgovarajući znak u ključu. Koristimo Vižnerovu tablicu za šifrovanje i dešifrovanje.

Vižnerova šifra

1. Pored poruke, potrebna nam je i **ključna reč**. Ona se zatim produži na sledeći način i dobijemo **ključ**:

ključna reč	KROV
ključ	KROVKROVKR
poruka	DOLAZISADA

2. Pri obradi svakog znaka u poruci koristimo posebnu monoalfabetsku supstituciju koju određuje odgovarajući znak u ključu. Koristimo Vižnerovu tablicu za šifrovanje i dešifrovanje.
3. Iz tablice se (proveriti!) dobije NFZVJZGVNR.

Vižnerova šifra

1. Pored poruke, potrebna nam je i **ključna reč**. Ona se zatim produži na sledeći način i dobijemo **ključ**:

ključna reč	KROV
ključ	KROVKROVKR
poruka	DOLAZISADA

2. Pri obradi svakog znaka u poruci koristimo posebnu monoalfabetsku supstituciju koju određuje odgovarajući znak u ključu. Koristimo Vižnerovu tablicu za šifrovanje i dešifrovanje.
3. Iz tablice se (proveriti!) dobije NFZVJZGVNR.
4. Moramo da znamo ključnu reč!

Vižnerov kvadrat

	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

Relacija kongruencije po modulu

- **Teorija brojeva** je značajna u kriptografiji, pre svega relacija **kongruencije po modulu** ("imaju isti ostatak pri deljenju"):

Relacija kongruencije po modulu

- **Teorija brojeva** je značajna u kriptografiji, pre svega relacija **kongruencije po modulu** ("imaju isti ostatak pri deljenju"):

Definicija

$$a \equiv b \pmod{m} \iff m \mid a - b$$

Relacija kongruencije po modulu

- **Teorija brojeva** je značajna u kriptografiji, pre svega relacija **kongruencije po modulu** ("imaju isti ostatak pri deljenju"):

Definicija

$$a \equiv b \pmod{m} \iff m \mid a - b$$

- $11 \equiv 2 \pmod{3}, 175 \equiv 0 \pmod{5}, 13 \equiv 17 \pmod{2} \ldots$

Relacija kongruencije po modulu

- **Teorija brojeva** je značajna u kriptografiji, pre svega relacija **kongruencije po modulu** ("imaju isti ostatak pri deljenju"):

Definicija

$$a \equiv b \pmod{m} \iff m \mid a - b$$

- $11 \equiv 2 \pmod{3}, 175 \equiv 0 \pmod{5}, 13 \equiv 17 \pmod{2} \ldots$
- Važna svojstva:

Relacija kongruencije po modulu

- **Teorija brojeva** je značajna u kriptografiji, pre svega relacija **kongruencije po modulu** ("imaju isti ostatak pri deljenju"):

Definicija

$$a \equiv b \pmod{m} \iff m \mid a - b$$

- $11 \equiv 2 \pmod{3}, 175 \equiv 0 \pmod{5}, 13 \equiv 17 \pmod{2} \ldots$
- Važna svojstva:
  1. $a \equiv b \pmod{m} \iff a \pm c \equiv b \pm c \pmod{m}$;

Relacija kongruencije po modulu

- **Teorija brojeva** je značajna u kriptografiji, pre svega relacija **kongruencije po modulu** ("imaju isti ostatak pri deljenju"):

Definicija

$$a \equiv b \pmod{m} \iff m \mid a - b$$

- $11 \equiv 2 \pmod{3}, 175 \equiv 0 \pmod{5}, 13 \equiv 17 \pmod{2} \ldots$
- Važna svojstva:
  1. $a \equiv b \pmod{m} \iff a \pm c \equiv b \pm c \pmod{m}$;
  2. $a \equiv b \pmod{m} \implies a \cdot c \equiv b \cdot c \pmod{m}$.

Relacija kongruencije po modulu

- **Teorija brojeva** je značajna u kriptografiji, pre svega relacija **kongruencije po modulu** ("imaju isti ostatak pri deljenju"):

Definicija

$$a \equiv b \pmod{m} \iff m \mid a - b$$

- $11 \equiv 2 \pmod{3}, 175 \equiv 0 \pmod{5}, 13 \equiv 17 \pmod{2} \ldots$
- Važna svojstva:
  1. $a \equiv b \pmod{m} \iff a \pm c \equiv b \pm c \pmod{m}$;
  2. $a \equiv b \pmod{m} \implies a \cdot c \equiv b \cdot c \pmod{m}$.
- Možemo da sabiramo, oduzimamo i množimo po nekom modulu: $(7 + 5) \mod 10 = 2, (7 \cdot 8) \mod 20 = 16$.

Eksponencijalne kongruencije

- Ako su a i m uzajamno prosti brojevi (NZD $(a, m) = 1$), tada posmatrajmo niz: a, a^2, a^3, …, a^k, …

Eksponencijalne kongruencije

- Ako su a i m uzajamno prosti brojevi (NZD $(a, m) = 1$), tada posmatrajmo niz: a, a^2, a^3, …, a^k, …
- Ovaj niz je periodičan:

Eksponencijalne kongruencije

- Ako su a i m uzajamno prosti brojevi (NZD $(a, m) = 1$), tada posmatrajmo niz: a, a^2, a^3, …, a^k, …
- Ovaj niz je periodičan:

Teorema (Ojlerova)

Ako su a i m uzajamno prosti prirodni brojevi, tada je:

$$a^{\varphi(m)} \equiv 1 \pmod{m}$$

gde je $\varphi(m)$ broj prirodnih brojeva manjih od m koji su sa njim uzajamno prosti (Ojlerova funkcija).

Eksponencijalne kongruencije

- Ako su a i m uzajamno prosti brojevi (NZD $(a, m) = 1$), tada posmatrajmo niz: a, a^2, a^3, …, a^k, …
- Ovaj niz je periodičan:

Teorema (Ojlerova)

Ako su a i m uzajamno prosti prirodni brojevi, tada je:

$$a^{\varphi(m)} \equiv 1 \pmod{m}$$

gde je $\varphi(m)$ broj prirodnih brojeva manjih od m koji su sa njim uzajamno prosti (Ojlerova funkcija).

- **Primer:** $\varphi(15) = 8$, $2^8 \equiv 256 \equiv 1 \pmod{15}$ jer $255 = 17 \cdot 15 + 1$.

Eksponencijalne kongruencije

- Ako su a i m uzajamno prosti brojevi (NZD $(a, m) = 1$), tada posmatrajmo niz: a, a^2, a^3, …, a^k, …
- Ovaj niz je periodičan:

Teorema (Ojlerova)

Ako su a i m uzajamno prosti prirodni brojevi, tada je:

$$a^{\varphi(m)} \equiv 1 \pmod{m}$$

gde je $\varphi(m)$ broj prirodnih brojeva manjih od m koji su sa njim uzajamno prosti (Ojlerova funkcija).

- **Primer:** $\varphi(15) = 8$, $2^8 \equiv 256 \equiv 1 \pmod{15}$ jer $255 = 17 \cdot 15 + 1$.
- Ali kako brzo izračunati $\varphi(n)$?

Faktorizacija broja

Teorema (Kanonska faktorizacija)

Svaki prirodan broj n veći od 1 može da se predstavi u obliku $n = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_k^{\alpha_k}$ za neki izbor prostih brojeva $p_1, p_2, \dots, p_k$ i neke nenegativne cele brojeve $\alpha_1, \alpha_2, \dots, \alpha_k$ na jedistven način (do na redosled činilaca).

Računanje vrednosti Ojlerove funkcije

Teorema

Za kanonsku faktorizaciju $n = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_k^{\alpha_k}$*:*

$$\varphi(n) = n \prod_{i=1}^{k} \left(1 - \frac{1}{p_i}\right)$$

Posledica

Ako su p *i* q *različiti prosti brojevi, tada:*

$$\varphi(pq) = (p-1)(q-1)$$

Multiplikativni inverzi i $\mathbb{Z}_n$

- Sada posmatrajmo skup $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$, pri čemu su sabiranje $+$ i množenje $\cdot$ ovde *zatvoreni* tako što se sve vrši "po modulu n". Tako je $3 \cdot 2 = 1$ u $\mathbb{Z}_5$.

Multiplikativni inverzi i $\mathbb{Z}_n$

- Sada posmatrajmo skup $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$, pri čemu su sabiranje $+$ i množenje $\cdot$ ovde *zatvoreni* tako što se sve vrši "po modulu n". Tako je $3 \cdot 2 = 1$ u $\mathbb{Z}_5$.
- Kod sabiranja imamo inverzni element (možemo da pišemo "$-x$" umesto $n - x$). Šta je sa množenjem?

Multiplikativni inverzi i $\mathbb{Z}_n$

- Sada posmatrajmo skup $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$, pri čemu su sabiranje $+$ i množenje $\cdot$ ovde *zatvoreni* tako što se sve vrši "po modulu n". Tako je $3 \cdot 2 = 1$ u $\mathbb{Z}_5$.
- Kod sabiranja imamo inverzni element (možemo da pišemo "$-x$" umesto $n - x$). Šta je sa množenjem?

Definicija (Multiplikativni inverz)

*Za $a \in \mathbb{Z}_n$ kažemo da ima **multiplikativni inverz** (po modulu n) $b \in \mathbb{Z}_n$ ukoliko $ab = 1$. Pišemo $b = a^{-1}$.*

Multiplikativni inverzi i $\mathbb{Z}_n$

- Sada posmatrajmo skup $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$, pri čemu su sabiranje $+$ i množenje $\cdot$ ovde *zatvoreni* tako što se sve vrši "po modulu n". Tako je $3 \cdot 2 = 1$ u $\mathbb{Z}_5$.
- Kod sabiranja imamo inverzni element (možemo da pišemo "$-x$" umesto $n - x$). Šta je sa množenjem?

Definicija (Multiplikativni inverz)

*Za $a \in \mathbb{Z}_n$ kažemo da ima **multiplikativni inverz** (po modulu n) $b \in \mathbb{Z}_n$ ukoliko $ab = 1$. Pišemo $b = a^{-1}$.*

- Ako je $a \in \mathbb{Z}_n$ uzajamno prost sa n, tada je $a^{\varphi(n)-1}$ multiplikativni inverz od a.

Koji su teški problemi?

- Za naredna dva zadatka ne znamo koliko su "rešivi", odnosno ne znamo da li su u $\mathcal{P}$: ne znamo da li postoje algoritmi polinomne složenosti koji ih rešavaju (gde se veličinom ulaza smatra broj bitova neophodan za predstavljanje broja).

Koji su teški problemi?

- Za naredna dva zadatka ne znamo koliko su "rešivi", odnosno ne znamo da li su u $\mathcal{P}$: ne znamo da li postoje algoritmi polinomne složenosti koji ih rešavaju (gde se veličinom ulaza smatra broj bitova neophodan za predstavljanje broja).
  - **Faktorizacija.** Naći kanonsku faktorizaciju datog prirodnog broja n.

Koji su teški problemi?

- Za naredna dva zadatka ne znamo koliko su "rešivi", odnosno ne znamo da li su u $\mathcal{P}$: ne znamo da li postoje algoritmi polinomne složenosti koji ih rešavaju (gde se veličinom ulaza smatra broj bitova neophodan za predstavljanje broja).
  - **Faktorizacija.** Naći kanonsku faktorizaciju datog prirodnog broja n.
  - **Diskretni logaritam.** Za date brojeve k, a i n, naći neko b takvo da je $b^k \equiv a \pmod{n}$.

Koji su teški problemi?

- Za naredna dva zadatka ne znamo koliko su "rešivi", odnosno ne znamo da li su u $\mathcal{P}$: ne znamo da li postoje algoritmi polinomne složenosti koji ih rešavaju (gde se veličinom ulaza smatra broj bitova neophodan za predstavljanje broja).
  - **Faktorizacija.** Naći kanonsku faktorizaciju datog prirodnog broja n.
  - **Diskretni logaritam.** Za date brojeve k, a i n, naći neko b takvo da je $b^k \equiv a \pmod{n}$.
- Zato smatramo da su savremeni algoritmi *pouzdani*.

Koji su teški problemi?

- Za naredna dva zadatka ne znamo koliko su "rešivi", odnosno ne znamo da li su u $\mathcal{P}$: ne znamo da li postoje algoritmi polinomne složenosti koji ih rešavaju (gde se veličinom ulaza smatra broj bitova neophodan za predstavljanje broja).
  - **Faktorizacija.** Naći kanonsku faktorizaciju datog prirodnog broja n.
  - **Diskretni logaritam.** Za date brojeve k, a i n, naći neko b takvo da je $b^k \equiv a \pmod{n}$.
- Zato smatramo da su savremeni algoritmi *pouzdani*.
- Priča je drugačija u domenu *kvantnog računarstva*...

Kerkhofsov princip

- Do sada smo bili u situaciji da, ako neko sazna koju šifru koristimo, nema nam spasa. Ali to je neizbežno!
- Moramo da pretpostavimo da *neprijatelji* i *špijuni* već znaju koju šifru koristimo.

Kerkhofsov princip

Sistem ne bi trebalo da bude tajan, i ne bi trebalo da ugrozi komunikaciju ako padne u ruke neprijatelja.

- Nasuprot tome, *security through obscurity*.

Jednokratne šifre

- Alisa i Bob žele da komuniciraju tako da im niko nikada ne otkrije ključ.
- Zato će u svakoj poruci menjati ključ, pri čemu će se unapred dogovoriti kada će koji koristiti. Poželjno je da ti ključevi budu nasumično generisani.
- Tako Eva i Melet, čak i da uspeju da izvrše dekripciju, moraće ponovo da prođu kroz taj silan posao (u zavisnosti od šifre koja se koristi).
- U idealnim okolnostima (tajnost je očuvana) sistem nema slabosti.

Primer jednokratne šifre

1. Svakom slovu dodelimo broj: `A`, `B`, `C`, ..., `Z` su, redom, 0, $1, 2, \ldots, 25$.

Primer jednokratne šifre

1. Svakom slovu dodelimo broj: `A`, `B`, `C`, `...`, `Z` su, redom, 0, $1, 2, \ldots, 25$.
2. Ključ je neki dugačak niz brojeva (odnosno znakova) manjih od 26 (vrednosti x_i) specifičan za ovu poruku.

Primer jednokratne šifre

1. Svakom slovu dodelimo broj: `A`, `B`, `C`, `...`, `Z` su, redom, 0, $1, 2, \ldots, 25$.
2. Ključ je neki dugačak niz brojeva (odnosno znakova) manjih od 26 (vrednosti x_i) specifičan za ovu poruku.
3. Izvorni tekst je, dakle, takođe neki niz brojeva (y_i).

Primer jednokratne šifre

1. Svakom slovu dodelimo broj: `A`, `B`, `C`, `...`, `Z` su, redom, 0, $1, 2, \ldots, 25$.
2. Ključ je neki dugačak niz brojeva (odnosno znakova) manjih od 26 (vrednosti x_i) specifičan za ovu poruku.
3. Izvorni tekst je, dakle, takođe neki niz brojeva (y_i).
4. Šifrat se dobija kao zbir odgovarajućih elemenata po modulu 26: $x_i + y_i \mod 26$.

`plaintext`	`HELLO`
`plaintext, brojevi`	`7, 4, 11, 11, 14`
`ključ`	`XYAMN`
`ključ, brojevi`	`23, 24, 0, 12, 13`
`šifrat, brojevi`	`4, 2, 11, 23, 1`
`šifrat`	`ECLXB`

ADFGVX, prvi korak

Nemačka šifra u Prvom svetskom ratu.

1. Koristimo fiksnu tablicu 6×6:

	A	D	F	G	V	X
A	k	z	w	r	1	f
D	9	b	6	c	l	5
F	q	7	j	p	g	x
G	e	v	y	3	a	n
V	8	o	d	h	0	2
X	u	4	i	s	t	m

ADFGVX, prvi korak

Nemačka šifra u Prvom svetskom ratu.

1. Koristimo fiksnu tablicu 6×6:

	A	D	F	G	V	X
A	k	z	w	r	1	f
D	9	b	6	c	l	5
F	q	7	j	p	g	x
G	e	v	y	3	a	n
V	8	o	d	h	0	2
X	u	4	i	s	t	m

2. Obrađujemo pojedinačne znake i čitamo zamene iz tabele: achtung $\rightarrow$ GV DG VG XV XA GX FV.

ADFGVX, drugi korak

1. Ovaj šifrat upisujemo u tablicu (bez ikakvog dodavanja znakova), pri čemu upisujemo u zaglavlje i ključ:

k	e	y	s
G	V	D	G
V	G	X	V
X	A	G	X
F	V		

ADFGVX, drugi korak

1. Ovaj šifrat upisujemo u tablicu (bez ikakvog dodavanja znakova), pri čemu upisujemo u zaglavlje i ključ:

k	e	y	s
G	V	D	G
V	G	X	V
X	A	G	X
F	V		

2. Kolone se ispermutuju tako da keys $\rightarrow$ eksy:

e	k	s	y
V	G	G	D
G	V	V	X
A	X	X	G
V	F		

ADFGVX, drugi korak

1. Ovaj šifrat upisujemo u tablicu (bez ikakvog dodavanja znakova), pri čemu upisujemo u zaglavlje i ključ:

k	e	y	s
G	V	D	G
V	G	X	V
X	A	G	X
F	V		

2. Kolone se ispermutuju tako da `keys` $\rightarrow$ `eksy`:

e	k	s	y
V	G	G	D
G	V	V	X
A	X	X	G
V	F		

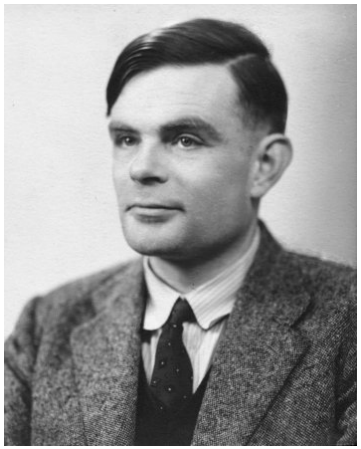
3. Šifrat se dobije čitanjem "odozgo na dole": `VGAV GVXF GVX DXG`.

Dalji razvoj šifara

- ADFGVX koristi samo jednu supstituciju i jednu transpoziciju, gde je zamena fiksna. Za pojačanu sigurnost: više iteracija!

Dalji razvoj šifara

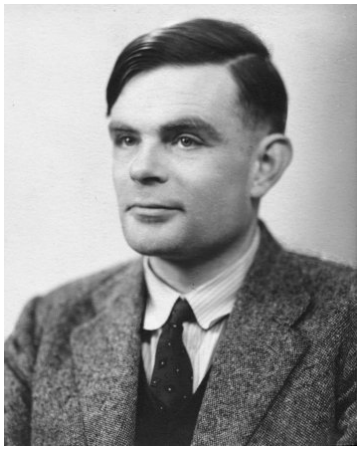
- ADFGVX koristi samo jednu supstituciju i jednu transpoziciju, gde je zamena fiksna. Za pojačanu sigurnost: više iteracija!
- Nemačka ENIGMA u Drugom svetskom ratu.



Slika: Alan Tjuring

Dalji razvoj šifara

- ADFGVX koristi samo jednu supstituciju i jednu transpoziciju, gde je zamena fiksna. Za pojačanu sigurnost: više iteracija!
- Nemačka ENIGMA u Drugom svetskom ratu.



Slika: Alan Tjuring

- Sa razvojem računara se pojavljuju novi koncepti i novi algoritmi. Do kraja XX veka je bio široko rasprostranjen DES (*Data Encryption System*).

Dve osnovne paradigme

- Savremena kriptografija zahteva da svrstamo većinu algoritama u jednu od naredne dve kategorije:

Dve osnovne paradigme

- Savremena kriptografija zahteva da svrstamo većinu algoritama u jednu od naredne dve kategorije:
  - **Shared key:** pošaljilac i primalac imaju **zajednički tajni ključ** koji se koristi za šifrovanje i dešifrovanje.

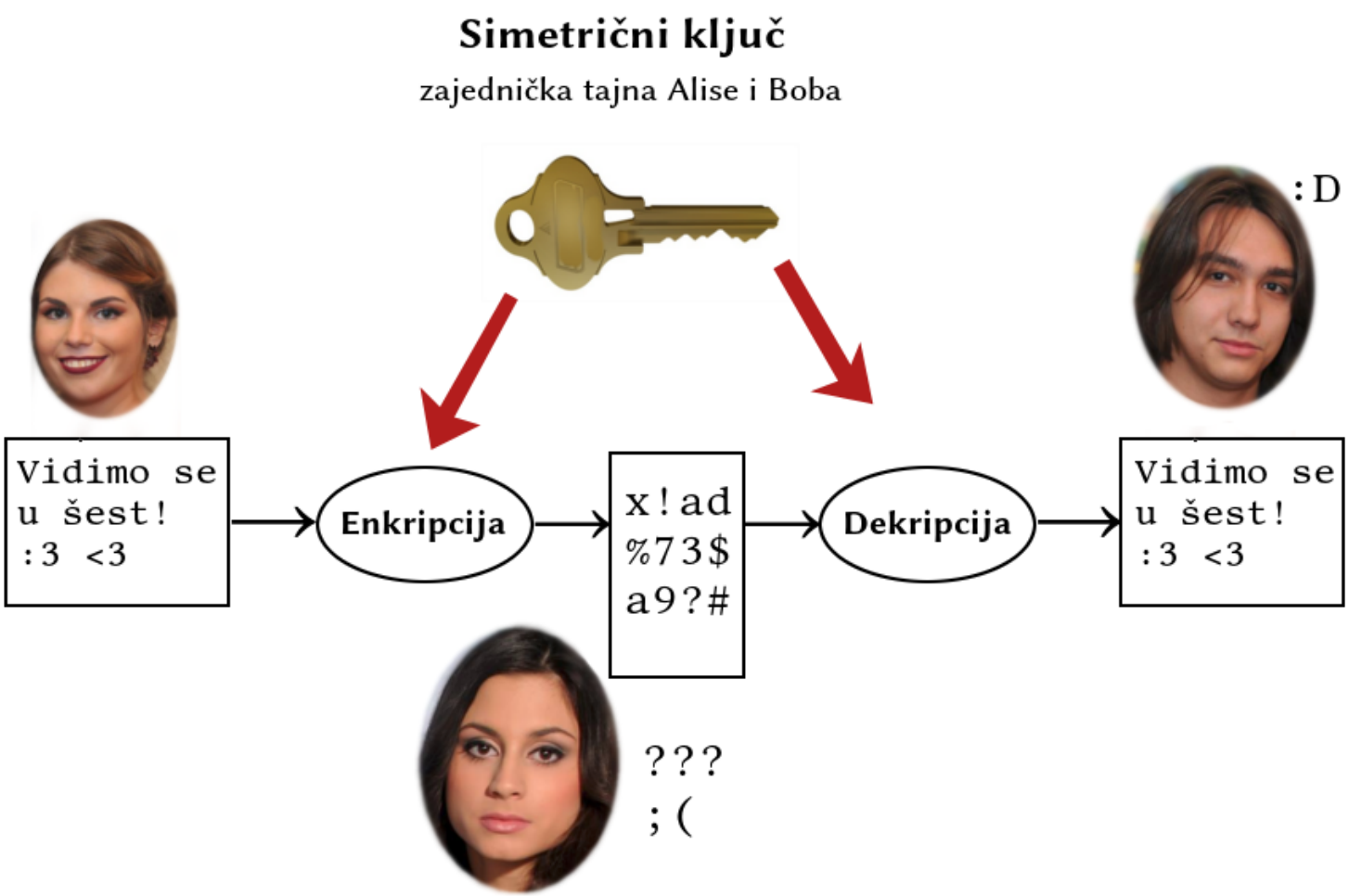
Dve osnovne paradigme

- Savremena kriptografija zahteva da svrstamo većinu algoritama u jednu od naredne dve kategorije:
  - **Shared key:** pošaljilac i primalac imaju **zajednički tajni ključ** koji se koristi za šifrovanje i dešifrovanje.
  - **Public key:** primalac objavljuje **javni ključ** i svi mogu da mu šalju poruke, ali jedino primalac može da dešifruje korišćenjem svog **privatnog ključa**.

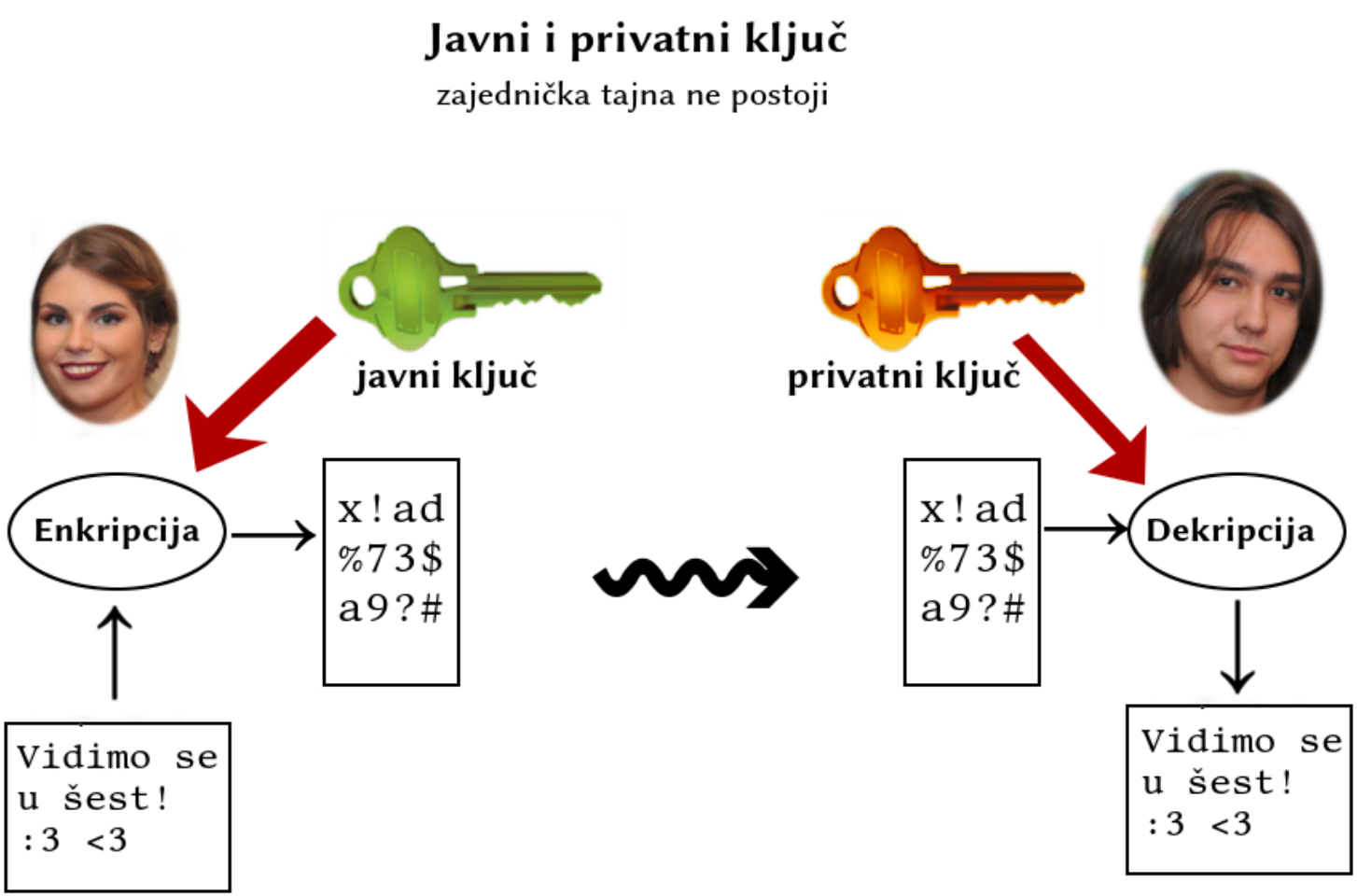
Dve osnovne paradigme

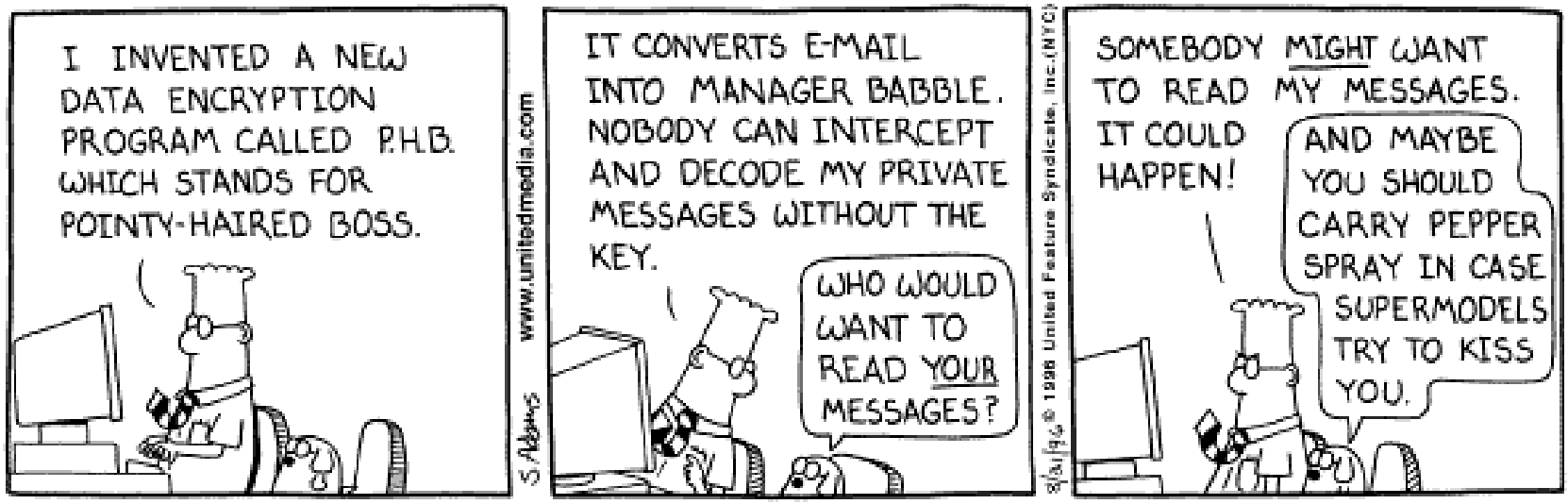
- Savremena kriptografija zahteva da svrstamo većinu algoritama u jednu od naredne dve kategorije:
  - **Shared key:** pošaljilac i primalac imaju **zajednički tajni ključ** koji se koristi za šifrovanje i dešifrovanje.
  - **Public key:** primalac objavljuje **javni ključ** i svi mogu da mu šalju poruke, ali jedino primalac može da dešifruje korišćenjem svog **privatnog ključa**.
- Videćemo nekoliko primera kasnije.

Shared key


Simetrični ključ
zajednička tajna Alise i Boba
:D
Vidimo se u šest! :3 <3
Enkripcija
x!ad %73$ a9?#
Dekripcija
Vidimo se u šest! :3 <3
???
;(

Public key


Javni i privatni ključ
zajednička tajna ne postoji
javni ključ
privatni ključ
Enkripcija
x!ad
%73$
a9?#
Vidimo se
u šest!
:3 <3
x!ad
%73$
a9?#
Dekripcija
Vidimo se
u šest!
:3 <3

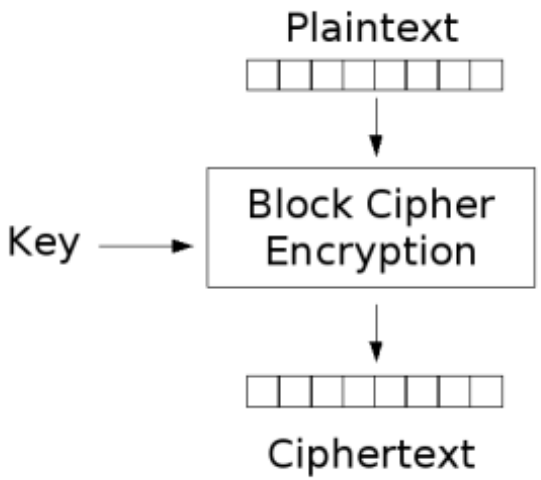


I frequently send myself encrypted messages, but I can never decrypt them so I don't know what I'm talking about.

A. C. Norman

Blok šifre

- Dele poruku na blokove (delove) i šifriraju blok po blok
- Šifrat je iste dužine kao i poruka
- Tpične dužine blokova su 64 i 128 bita



Slika: Blok šifra

Feistel-ova struktura blokovskih šifara

- Feistel je predstavio algoritam spajanja dve ili više jednostavnih šifri u sekvencu tako da je krajnja šifra dosta jača
- Primenjuje koncept difuzije (*diffusion*) i konfuzije (*confusion*)
- Primenjuje se u mnogim šiframa danas
- Pristup:
  - Poruka se deli na dva dela
  - Podključevi su generisani iz glavnog ključa
  - Funkcija, F, primenjuje se na desnu polovinu poruke
  - Primenjuje se metod zamene na levu polovinu (ovde XOR-ovanje)
  - Permutacija (ovde zamena leve i desne polovine poruke)

Difuzija i konfuzija

Difuzija

- Neutrališe statističke karakteristike poruke

Konfuzija

Difuzija i konfuzija

Difuzija

- Neutrališe statističke karakteristike poruke
- Postiže se tako što svaki bit poruke utiče na više bitova šifrata

Konfuzija

Difuzija i konfuzija

Difuzija

- Neutrališe statističke karakteristike poruke
- Postiže se tako što svaki bit poruke utiče na više bitova šifrata
- Kako? - Nekoliko permutacija (traspozicija) zaredom uz primenu određene funkcije nakon svake permutacije

Konfuzija

Difuzija i konfuzija

Difuzija

- Neutrališe statističke karakteristike poruke
- Postiže se tako što svaki bit poruke utiče na više bitova šifrata
- Kako? - Nekoliko permutacija (traspozicija) zaredom uz primenu određene funkcije nakon svake permutacije

Konfuzija

- Stvara što komplikovaniju vezu između šifrata i ključa

Difuzija i konfuzija

Difuzija

- Neutrališe statističke karakteristike poruke
- Postiže se tako što svaki bit poruke utiče na više bitova šifrata
- Kako? - Nekoliko permutacija (traspozicija) zaredom uz primenu određene funkcije nakon svake permutacije

Konfuzija

- Stvara što komplikovaniju vezu između šifrata i ključa
- Čak i da Eva ili Melet mogu naći neke statističke karakteristike, i dalje teško nalaze ključ

Difuzija i konfuzija

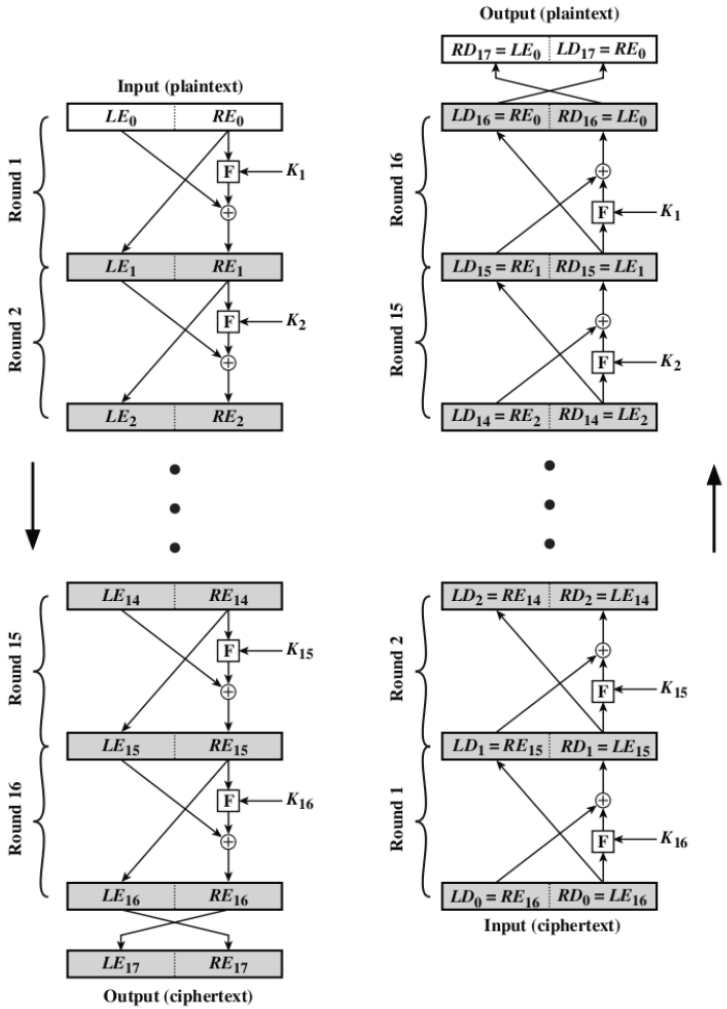
Difuzija

- Neutrališe statističke karakteristike poruke
- Postiže se tako što svaki bit poruke utiče na više bitova šifrata
- Kako? - Nekoliko permutacija (traspozicija) zaredom uz primenu određene funkcije nakon svake permutacije

Konfuzija

- Stvara što komplikovaniju vezu između šifrata i ključa
- Čak i da Eva ili Melet mogu naći neke statističke karakteristike, i dalje teško nalaze ključ
- Kako? - Primenom kompleksnog algoritma zamene

Feistel-ova enkripcija i dekripcija



Još malo o Feistel-ovoj strukturi

- Sama implementacija zavisi od različitih karakteristika strukture:

Još malo o Feistel-ovoj strukturi

- Sama implementacija zavisi od različitih karakteristika strukture:
  - Dužina bloka (64, 128 bita) - veće vrednosti dovode do veće difuzije

Još malo o Feistel-ovoj strukturi

- Sama implementacija zavisi od različitih karakteristika strukture:
  - Dužina bloka (64, 128 bita) - veće vrednosti dovode do veće difuzije
  - Dužina ključa - veće vrednosti dovode do veće konfuzije i veće otpornosti na brute force napade

Još malo o Feistel-ovoj strukturi

- Sama implementacija zavisi od različitih karakteristika strukture:
  - Dužina bloka (64, 128 bita) - veće vrednosti dovode do veće difuzije
  - Dužina ključa - veće vrednosti dovode do veće konfuzije i veće otpornosti na brute force napade
  - Broj rundi

Još malo o Feistel-ovoj strukturi

- Sama implementacija zavisi od različitih karakteristika strukture:
  - Dužina bloka (64, 128 bita) - veće vrednosti dovode do veće difuzije
  - Dužina ključa - veće vrednosti dovode do veće konfuzije i veće otpornosti na brute force napade
  - Broj rundi
  - Algoritam za generisanje podključeva - što kompleksniji

Još malo o Feistel-ovoj strukturi

- Sama implementacija zavisi od različitih karakteristika strukture:
  - Dužina bloka (64, 128 bita) - veće vrednosti dovode do veće difuzije
  - Dužina ključa - veće vrednosti dovode do veće konfuzije i veće otpornosti na brute force napade
  - Broj rundi
  - Algoritam za generisanje podključeva - što kompleksniji
  - Funkcija F - takođe što kompleksnija

Još malo o Feistel-ovoj strukturi

- Sama implementacija zavisi od različitih karakteristika strukture:
  - Dužina bloka (64, 128 bita) - veće vrednosti dovode do veće difuzije
  - Dužina ključa - veće vrednosti dovode do veće konfuzije i veće otpornosti na brute force napade
  - Broj rundi
  - Algoritam za generisanje podključeva - što kompleksniji
  - Funkcija F - takođe što kompleksnija
- Ostali faktori koji utiču su brzina enkripcije i težina analize

Šta je DES?

- Najznačajniji algoritam simetrične kriptografije XX veka

Šta je DES?

- Najznačajniji algoritam simetrične kriptografije XX veka
- Veoma doprineo razvitku moderne kriptografije

Šta je DES?

- Najznačajniji algoritam simetrične kriptografije XX veka
- Veoma doprineo razvitku moderne kriptografije
- Predstavlja blokovski tip šifre

Šta je DES?

- Najznačajniji algoritam simetrične kriptografije XX veka
- Veoma doprineo razvitku moderne kriptografije
- Predstavlja blokovski tip šifre
- Koristi 56-bitni ključ i 64-bitne blokove

Šta je DES?

- Najznačajniji algoritam simetrične kriptografije XX veka
- Veoma doprineo razvitku moderne kriptografije
- Predstavlja blokovski tip šifre
- Koristi 56-bitni ključ i 64-bitne blokove
- Baziran na Feistel-ovoj strukturi sa 16 rundi

Šta je DES?

- Najznačajniji algoritam simetrične kriptografije XX veka
- Veoma doprineo razvitku moderne kriptografije
- Predstavlja blokovski tip šifre
- Koristi 56-bitni ključ i 64-bitne blokove
- Baziran na Feistel-ovoj strukturi sa 16 rundi
- Više se ne koristi (tehnologija ga je nadživela :)) ali su neka njegova unapređenja i dalje u upotrebi

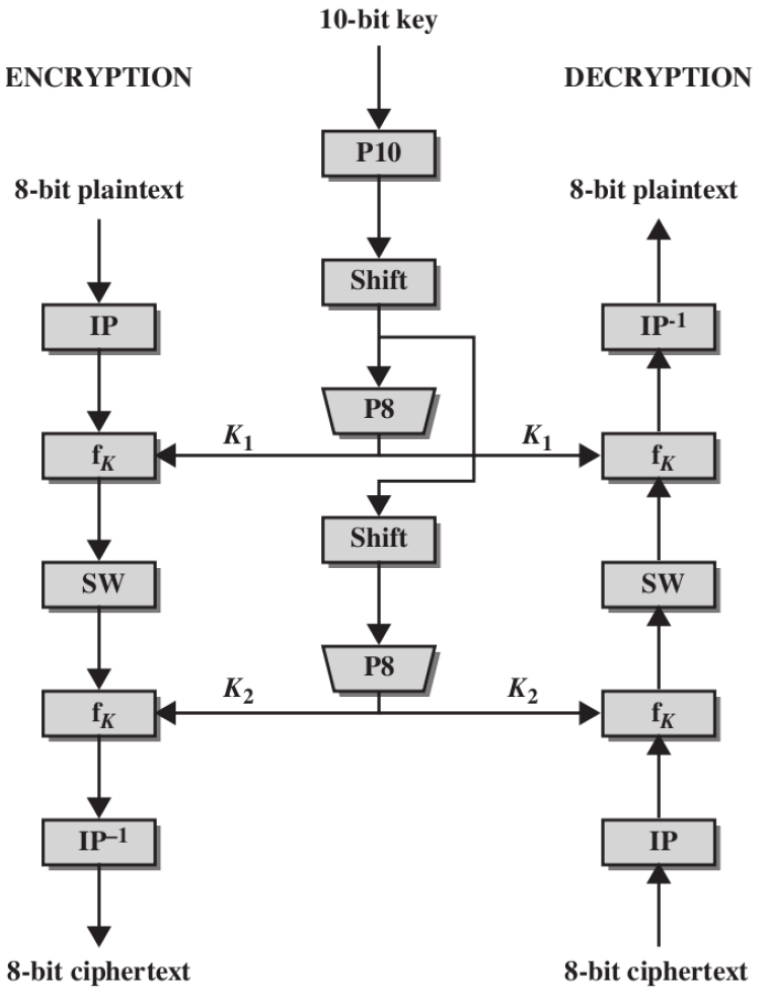
Kratka istorija

- 1973. NIST (tada NBS) izdaje obaveštenje da traži šifarski algoritam koji bi postao standard
- IBM je već razvio algoritam LUCIFER koji je koristio 64-bitne blokove i 128-bitni ključ
- 1974. ga predaje NIST-u
- LUCIFER biva pregledan i ponovno dizajniran od strane NSA
- 1977. usvojena modifikacija i nazvana DES

S-DES - Simplified DES

- Pojednostavljeni DES, pogodan za demonstraciju
- Ne koristi se, dizajniran u edukacione svrhe
- Ulazni blok: 8 bita
- Izlazni blok: 8 bita
- Ključ: 10 bita
- Rundi: 2
- Podključevi generisani permutacijama i pomeranjima ulevo
- Enkripcija: inicijalna permutacija, funkcija, zamena polovina
- Dekripcija: Ista kao enkripcija, samo se ključevi koriste u obrnutom smeru

S-DES algoritam



S-DES operacije 1

- P10 (permutovanje)

  Ulaz : 1 2 3 4 5 6 7 8 9 10
  Izlaz: 3 5 2 7 4 10 1 9 8 6

- P8 (selekcija i permutovanje)

  Ulaz : 1 2 3 4 5 6 7 8 9 10
  Izlaz: 6 3 7 4 8 5 10 9

- P4 (permutovanje)

  Ulaz : 1 2 3 4
  Izlaz: 2 4 3 1

S-DES operacije 2

- EP (proširenje i permutovanje)

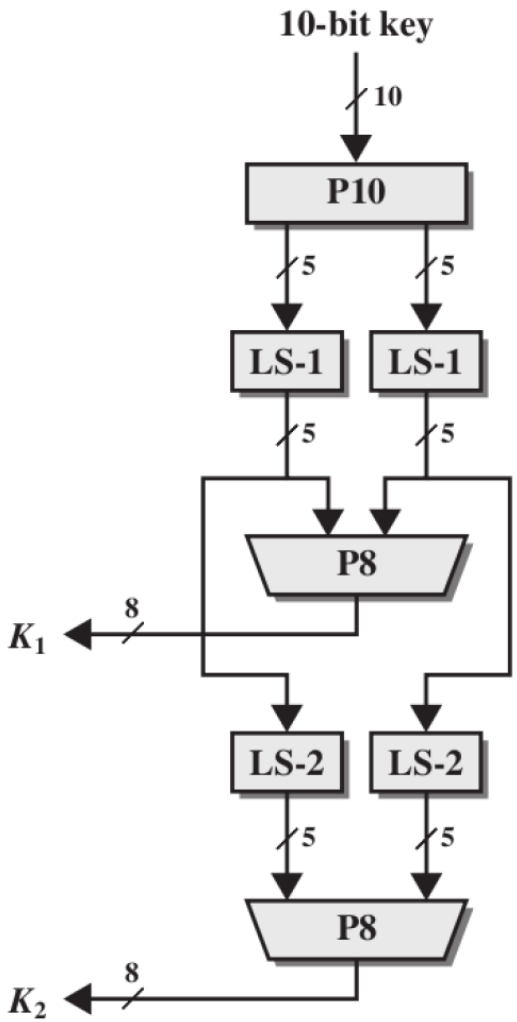
Ulaz :	1	2	3	4				
Izlaz:	4	1	2	3	2	3	4	1

- IP (inicijalna permutacija)

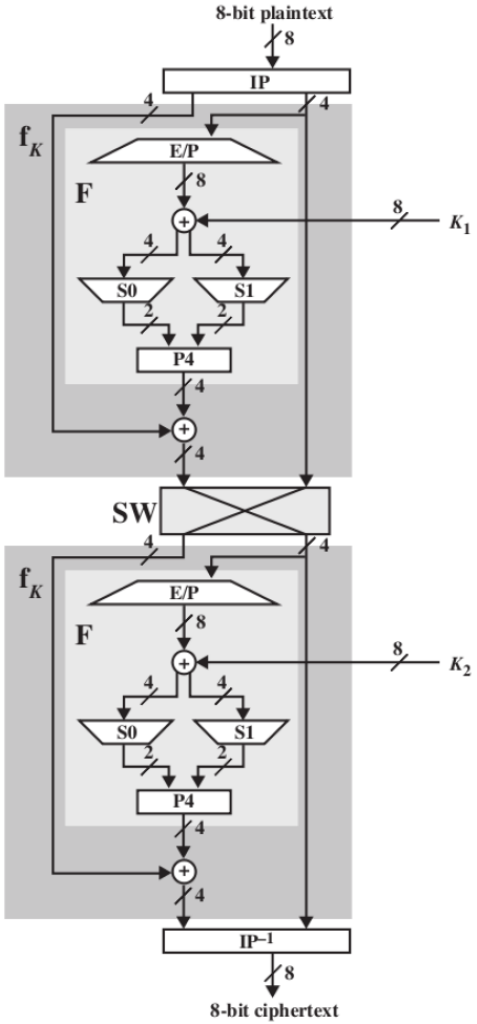
Ulaz :	1	2	3	4	5	6	7	8
Izlaz:	2	6	3	1	4	8	5	7

- IP^{-1} (inverz IP-a)
- LS-1 (pomeranje ulevo za 1 poziciju)
- LS-2 (pomeranje ulevo za 2 pozicije)

S-DES generisanje ključa



S-DES Detalji enkripcije



S-DES S-kutije

- S-DES (i DES) vrše zamene pomoću S-kutija
- S-kutija je matrica
- Ulaz se koristi da se odrede vrsta i kolona, a izlaz je vrednost na tom mestu u matrici
- 4-bitni ulaz: $bit_1, bit_2, bit_3, bit_4$
- $bit_1 bit_4$ određuju vrstu (0, 1, 2 ili 3)
- $bit_2 bit_3$ određuju kolonu
- 2-bitni izlaz

$$S0 = \begin{bmatrix} 01 & 00 & 11 & 10 \\ 11 & 10 & 01 & 00 \\ 00 & 10 & 01 & 11 \\ 11 & 01 & 11 & 10 \end{bmatrix} \quad S1 = \begin{bmatrix} 00 & 01 & 10 & 11 \\ 10 & 00 & 01 & 11 \\ 11 & 00 & 01 & 00 \\ 10 & 01 & 00 & 11 \end{bmatrix}$$

S-DES zaključak

- S-DES izrazi:

$$ciphertext = IP^{-1}\left(f_{K_2}\left(SW\left(f_{K_1}\left(IP\left(plaintext\right)\right)\right)\right)\right)$$

$$plaintext = IP^{-1}\left(f_{K_1}\left(SW\left(f_{K_2}\left(IP\left(ciphertext\right)\right)\right)\right)\right)$$

- Sigurnost S-DES:
  - 10-bitni ključ $\rightarrow$ 1024 ključeva $\rightarrow$ brute force bez problema

Poređenje DES-a i S-DES-a

S-DES:

- 8-bitni blokovi
- 10-bit ključ
- IP: 8-bitna
- F radi sa 4 bita
- 2 S-kutije
- 2 runde

DES:

- 64-bitni blokovi
- 56-bitni ključ
- IP: 64 bita
- F radi na 32 bita
- 8 S-kutija
- 16 rundi

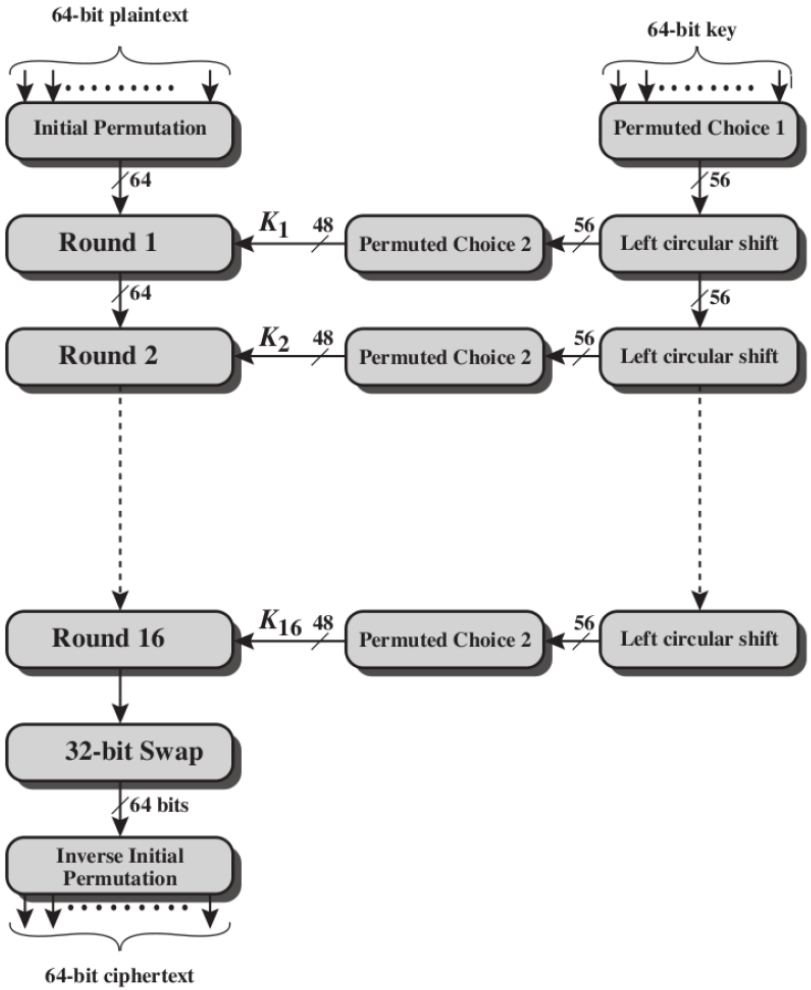
S-DES enkripcija:

$$ciphertext = IP^{-1}\left(f_{K_2}\left(SW\left(f_{K_1}\left(IP\left(plaintext\right)\right)\right)\right)\right)$$

DES enkripcija:

$$ciphertext = IP^{-1}\left(f_{K_{16}}\left(SW\left(f_{K_{15}}\left(SW\left(\ldots\left(f_{K_1}\left(IP\left(plaintext\right)\right)\right)\right)\right)\right)\right)\right)$$

DES algoritam



Tabele permutacija DES-a 1

IP							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

IP^{-1}							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Slika: Tabele inicijalne i inverzne inicijalne permutacije

Tabele permutacija DES-a 2

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

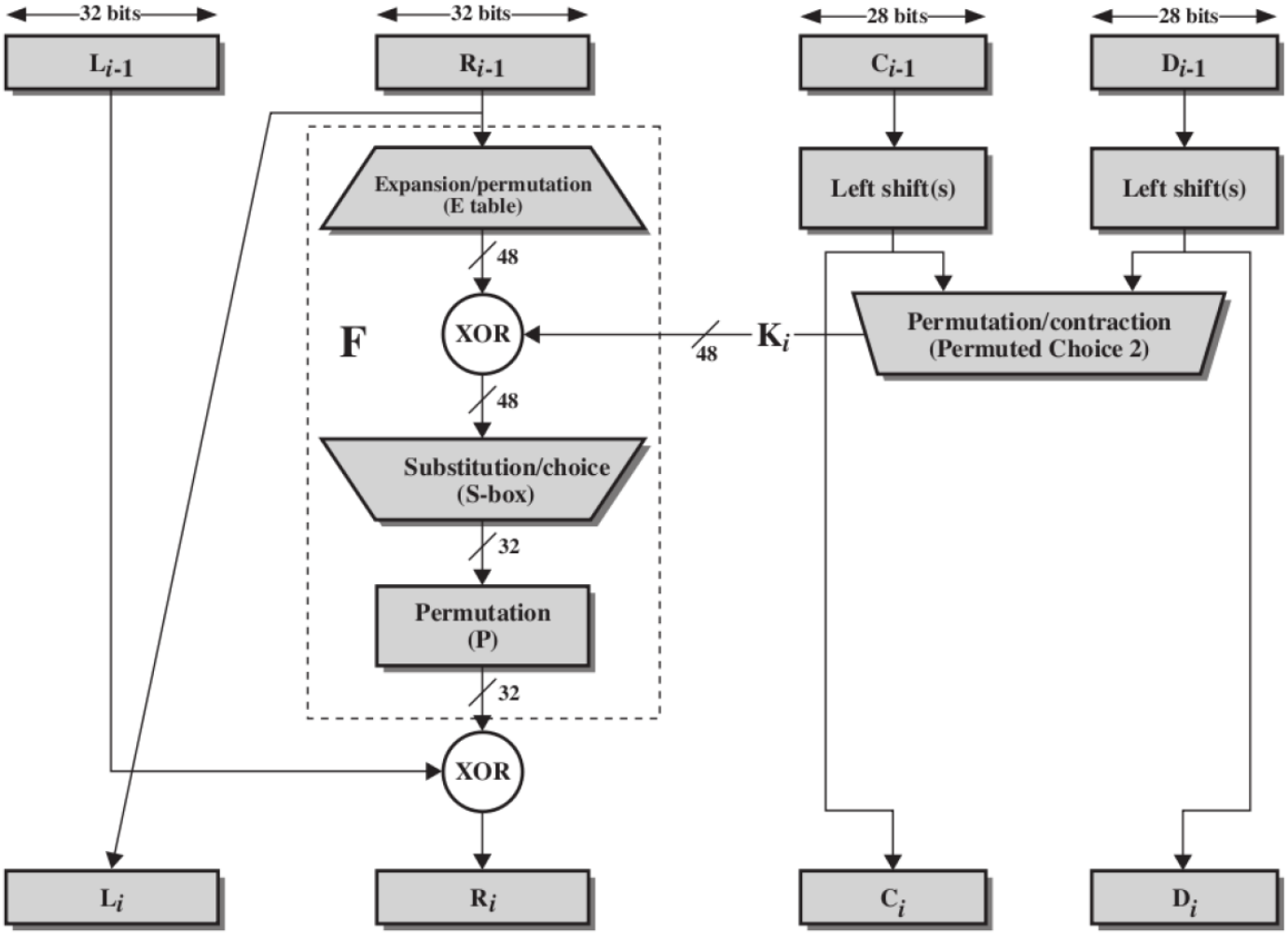
E

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

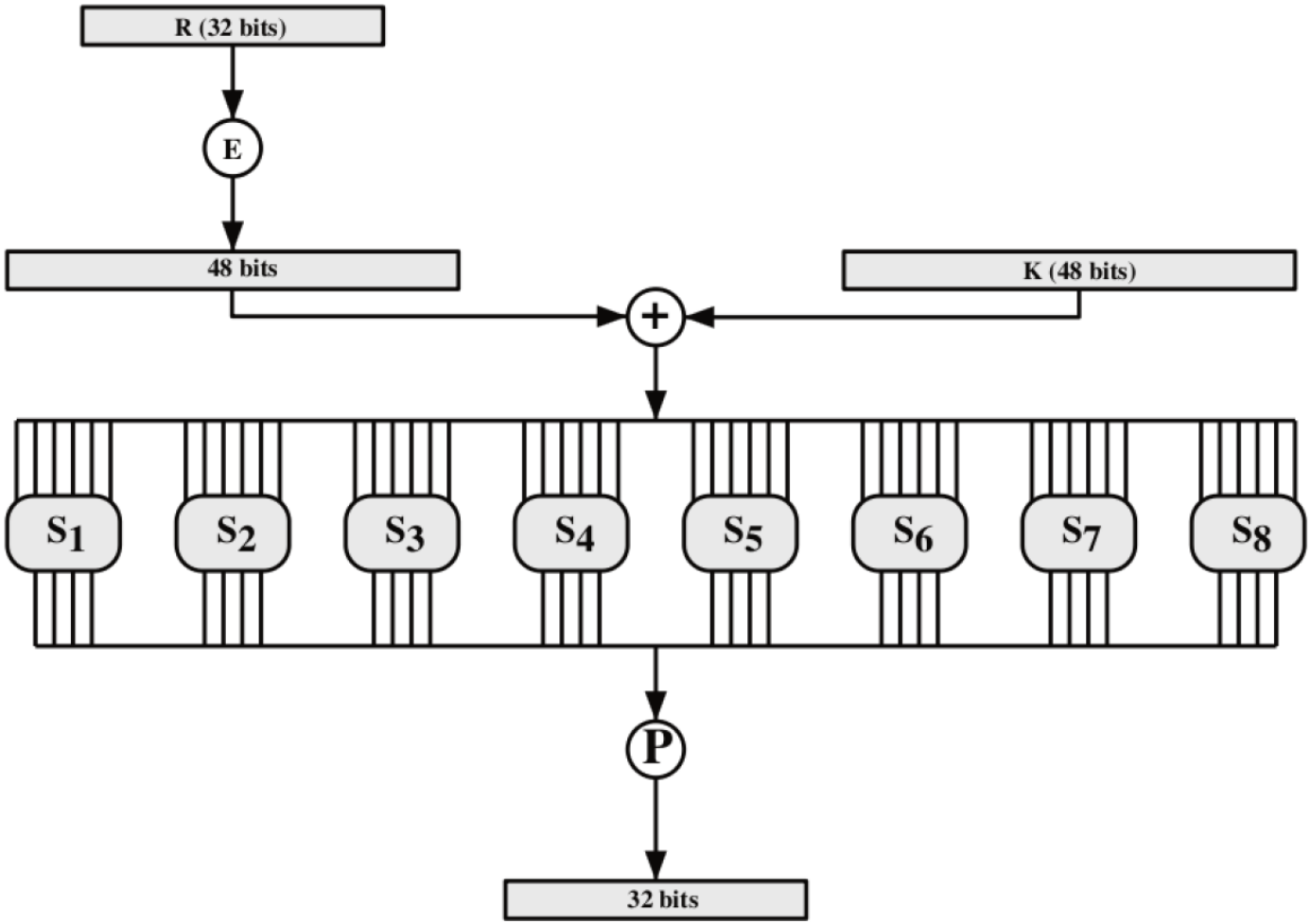
P

Slika: Tabele proširenja (E) i permutacije (P)

Jedna runda DES-a



Funkcija $F(R, K)$



S-kutije 1

S_1

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_2

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S-kutije 2

S_5

2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6

12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S_7

4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Generisanje podključeva

(a) Input Key

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

(b) Permuted Choice One (PC-1)

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

(c) Permuted Choice Two (PC-2)

14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

Alternative

- DES se može probiti brute force napadom

Alternative

- DES se može probiti brute force napadom
- Alternativna blok šifra koja koristi DES: višestruka enkripcija sa različitim ključevima

Alternative

- DES se može probiti brute force napadom
- Alternativna blok šifra koja koristi DES: višestruka enkripcija sa različitim ključevima
- Opcije:

Alternative

- DES se može probiti brute force napadom
- Alternativna blok šifra koja koristi DES: višestruka enkripcija sa različitim ključevima
- Opcije:
  1. Dupli DES: Iako bi teoretski trebalo da bude 2^{112} mogućih ključeva, ovakav sistem ima sigurnosni nivo 57-bitnog ključa

Alternative

- DES se može probiti brute force napadom
- Alternativna blok šifra koja koristi DES: višestruka enkripcija sa različitim ključevima
- Opcije:
  1. Dupli DES: Iako bi teoretski trebalo da bude 2^{112} mogućih ključeva, ovakav sistem ima sigurnosni nivo 57-bitnog ključa
  2. Trostruki DES (3DES) sa 2 ključa: brute force 2^{112}

Alternative

- DES se može probiti brute force napadom
- Alternativna blok šifra koja koristi DES: višestruka enkripcija sa različitim ključevima
- Opcije:
  1. Dupli DES: Iako bi teoretski trebalo da bude 2^{112} mogućih ključeva, ovakav sistem ima sigurnosni nivo 57-bitnog ključa
  2. Trostruki DES (3DES) sa 2 ključa: brute force 2^{112}
  3. Trostruki DES sa 3 ključa: brute force 2^{168}

Advanced Encryption Standard

- NIST 1997. kreće u potragu za novim standardom

Advanced Encryption Standard

- NIST 1997. kreće u potragu za novim standardom
  - Ciljevi: sigurnost, efikasna softverska/hardverska implementacija, malo memorije, paralelno procesiranje

Advanced Encryption Standard

- NIST 1997. kreće u potragu za novim standardom
  - Ciljevi: sigurnost, efikasna softverska/hardverska implementacija, malo memorije, paralelno procesiranje
  - Dosta kandidata (algoritama) iz celog sveta

Advanced Encryption Standard

- NIST 1997. kreće u potragu za novim standardom
  - Ciljevi: sigurnost, efikasna softverska/hardverska implementacija, malo memorije, paralelno procesiranje
  - Dosta kandidata (algoritama) iz celog sveta
  - Rijndael pobeđuje, 2001. postaje novi standard (AES)

Advanced Encryption Standard

- NIST 1997. kreće u potragu za novim standardom
  - Ciljevi: sigurnost, efikasna softverska/hardverska implementacija, malo memorije, paralelno procesiranje
  - Dosta kandidata (algoritama) iz celog sveta
  - Rijndael pobeđuje, 2001. postaje novi standard (AES)
- AES:

Advanced Encryption Standard

- NIST 1997. kreće u potragu za novim standardom
  - Ciljevi: sigurnost, efikasna softverska/hardverska implementacija, malo memorije, paralelno procesiranje
  - Dosta kandidata (algoritama) iz celog sveta
  - Rijndael pobeđuje, 2001. postaje novi standard (AES)
- AES:
  - Dužina bloka: 128 bita (moguće i druge dužine)

Advanced Encryption Standard

- NIST 1997. kreće u potragu za novim standardom
  - Ciljevi: sigurnost, efikasna softverska/hardverska implementacija, malo memorije, paralelno procesiranje
  - Dosta kandidata (algoritama) iz celog sveta
  - Rijndael pobeđuje, 2001. postaje novi standard (AES)
- AES:
  - Dužina bloka: 128 bita (moguće i druge dužine)
  - Dužina ključa: 128, 192 ili 256 bitova

Advanced Encryption Standard

- NIST 1997. kreće u potragu za novim standardom
  - Ciljevi: sigurnost, efikasna softverska/hardverska implementacija, malo memorije, paralelno procesiranje
  - Dosta kandidata (algoritama) iz celog sveta
  - Rijndael pobeđuje, 2001. postaje novi standard (AES)
- AES:
  - Dužina bloka: 128 bita (moguće i druge dužine)
  - Dužina ključa: 128, 192 ili 256 bitova
  - Rounds: 10, 12 ili 14 (zavisi od ključa)

Advanced Encryption Standard

- NIST 1997. kreće u potragu za novim standardom
  - Ciljevi: sigurnost, efikasna softverska/hardverska implementacija, malo memorije, paralelno procesiranje
  - Dosta kandidata (algoritama) iz celog sveta
  - Rijndael pobeđuje, 2001. postaje novi standard (AES)
- AES:
  - Dužina bloka: 128 bita (moguće i druge dužine)
  - Dužina ključa: 128, 192 ili 256 bitova
  - Rounds: 10, 12 ili 14 (zavisi od ključa)
  - Operations: XOR sa podključem, zamena korišćenjem S-kutija, mešanje korišćenjem aritmetike Galois-ovih polja

Advanced Encryption Standard

- NIST 1997. kreće u potragu za novim standardom
  - Ciljevi: sigurnost, efikasna softverska/hardverska implementacija, malo memorije, paralelno procesiranje
  - Dosta kandidata (algoritama) iz celog sveta
  - Rijndael pobeđuje, 2001. postaje novi standard (AES)
- AES:
  - Dužina bloka: 128 bita (moguće i druge dužine)
  - Dužina ključa: 128, 192 ili 256 bitova
  - Rounds: 10, 12 ili 14 (zavisi od ključa)
  - Operations: XOR sa podključem, zamena korišćenjem S-kutija, mešanje korišćenjem aritmetike Galois-ovih polja
- Široko rasprostranjen

Advanced Encryption Standard

- NIST 1997. kreće u potragu za novim standardom
  - Ciljevi: sigurnost, efikasna softverska/hardverska implementacija, malo memorije, paralelno procesiranje
  - Dosta kandidata (algoritama) iz celog sveta
  - Rijndael pobeđuje, 2001. postaje novi standard (AES)
- AES:
  - Dužina bloka: 128 bita (moguće i druge dužine)
  - Dužina ključa: 128, 192 ili 256 bitova
  - Rounds: 10, 12 ili 14 (zavisi od ključa)
  - Operations: XOR sa podključem, zamena korišćenjem S-kutija, mešanje korišćenjem aritmetike Galois-ovih polja
- Široko rasprostranjen
- Smatra se sigurnim

Još neki simetrični algoritmi

- Blowfish (Schneier, 1993)
- Twofish (Schneier et al, 1998)
- Serpent (Anderson et al, 1998)
- Camellia (Mitsubishi/NTT, 2000)
- IDEA (Lai and Massey, 1991)
- CAST-128 (Adams and Tavares, 1996)
- CAST-256 (Adams and Tavares, 1998)
- RC5 (Rivest, 1994)
- RC6 (Rivest et al, 1998)

O algoritmu

- Algoritam za razmenu tajnog ključa

O algoritmu

- Algoritam za razmenu tajnog ključa
- Predložili ga Diffie i Hellman 1976. zajedno sa konceptima javnog ključa

O algoritmu

- Algoritam za razmenu tajnog ključa
- Predložili ga Diffie i Hellman 1976. zajedno sa konceptima javnog ključa
- Zasnovan na diskretnim logaritmima

O algoritmu

- Algoritam za razmenu tajnog ključa
- Predložili ga Diffie i Hellman 1976. zajedno sa konceptima javnog ključa
- Zasnovan na diskretnim logaritmima
- Sigurnost zasnovana na težini rešavanja diskretnog algoritma (slično kao faktorizacija kod RSA)

Kako radi?

1. Bob (ili Alisa) bira veliki prosti broj p i primitivni koren a $(\mathrm{mod}\ p)$. Oba broja se postavljaju javnim.

Kako radi?

1. Bob (ili Alisa) bira veliki prosti broj p i primitivni koren $a \pmod{p}$. Oba broja se postavljaju javnim.
2. Alisa nasumično bira tajni broj x, $(1 \leq x \leq p-2)$. Bob, takođe, nasumično bira tajni broj y, $(1 \leq y \leq p-2)$.

Kako radi?

1. Bob (ili Alisa) bira veliki prosti broj p i primitivni koren $a \pmod{p}$. Oba broja se postavljaju javnim.
2. Alisa nasumično bira tajni broj x, $(1 \leq x \leq p-2)$. Bob, takođe, nasumično bira tajni broj y, $(1 \leq y \leq p-2)$.
3. Alisa šalje Bobu $a^x \pmod{p}$, dok Bob njoj šalje $a^y \pmod{p}$

Kako radi?

1. Bob (ili Alisa) bira veliki prosti broj p i primitivni koren $a \pmod{p}$. Oba broja se postavljaju javnim.
2. Alisa nasumično bira tajni broj x, $(1 \leq x \leq p-2)$. Bob, takođe, nasumično bira tajni broj y, $(1 \leq y \leq p-2)$.
3. Alisa šalje Bobu $a^x \pmod{p}$, dok Bob njoj šalje $a^y \pmod{p}$
4. Koristeći dobivene poruke, svako od njih može da izračuna ključ K. Alisa računa K preko $K \equiv (a^y)^x \pmod{p}$, a Bob preko $K \equiv (a^y)^x \pmod{p}$

Uslovi kriptografije sa javnim ključem

1. Bobu je lako da generiše par (PU_b, PR_b)

Uslovi kriptografije sa javnim ključem

1. Bobu je lako da generiše par (PU_b, PR_b)
2. Lako je za Alisu, znajući PU_b i poruku M, da generiše šifrat:

$$C = \mathrm{E}(PU_b, M)$$

Uslovi kriptografije sa javnim ključem

1. Bobu je lako da generiše par (PU_b,PR_b)
2. Lako je za Alisu, znajući PU_b i poruku M, da generiše šifrat:

$$C = \mathrm{E}(PU_b, M)$$

3. Koristeći PR_b, Bob lako dešifruje kodiranu poruku:

$$M = \mathrm{D}(PR_b, C) = \mathrm{D}[PR_b, \mathrm{E}(PU_b, M)]$$

Uslovi kriptografije sa javnim ključem

1. Bobu je lako da generiše par (PU_b,PR_b)
2. Lako je za Alisu, znajući PU_b i poruku M, da generiše šifrat:

$$C = \mathrm{E}(PU_b, M)$$

3. Koristeći PR_b, Bob lako dešifruje kodiranu poruku:

$$M = \mathrm{D}(PR_b, C) = \mathrm{D}[PR_b, \mathrm{E}(PU_b, M)]$$

4. Evi ili Meletu je računski neizvodljivo da, znajući PU_b i C, odredi PR_b

Uslovi kriptografije sa javnim ključem

1. Bobu je lako da generiše par (PU_b,PR_b)
2. Lako je za Alisu, znajući PU_b i poruku M, da generiše šifrat:

$$C = \mathrm{E}(PU_b, M)$$

3. Koristeći PR_b, Bob lako dešifruje kodiranu poruku:

$$M = \mathrm{D}(PR_b, C) = \mathrm{D}[PR_b, \mathrm{E}(PU_b, M)]$$

4. Evi ili Meletu je računski neizvodljivo da, znajući PU_b i C, odredi PR_b
5. Evi ili Meletu je računski neizvodljivo da, znajući PU_b i C, odredi M

Uslovi kriptografije sa javnim ključem

1. Bobu je lako da generiše par (PU_b,PR_b)
2. Lako je za Alisu, znajući PU_b i poruku M, da generiše šifrat:
$$C = \mathrm{E}(PU_b, M)$$
3. Koristeći PR_b, Bob lako dešifruje kodiranu poruku:
$$M = \mathrm{D}(PR_b, C) = \mathrm{D}[PR_b, \mathrm{E}(PU_b, M)]$$
4. Evi ili Meletu je računski neizvodljivo da, znajući PU_b i C, odredi PR_b
5. Evi ili Meletu je računski neizvodljivo da, znajući PU_b i C, odredi M
6. (Opciono) Dva ključa mogu primenjena u proizvoljnom redosledu:
$$M = \mathrm{D}[PU_b, \mathrm{E}(PR_b, M)] = \mathrm{D}[PR_b, \mathrm{E}(PU_b, M)]$$

Šta je RSA?

- Ron Rivest, Adi Shamir i Len Adleman osnivači
- Kreiran 1977. na MIT-u
- Najpoznatiji i najkorišćeniji algoritam sa javnim ključem
- Blok šifra
- Poruke i šifrati su prirodni brojevi

RSA algoritam

Određivanje ključa

1. Nasumično se biraju prosti brojevi p i q i računa se $n = pq$
2. Zatim se selektuje e tako da važi:
   $\text{NZD}(\varphi(n), e) = 1, 1 < e < \varphi(n)$
3. Pronalazi se d takvo da je $d \equiv e^{-1} \pmod{\varphi(n)}$

$PU = \{e, n\}$, $PR = \{d, n\}$, p i q takođe privatni

Enkripcija

Enkripcija poruke M, gde je $M < n$:

$$C = M^e \bmod n$$

RSA algoritam

Dekripcija

Dekripcija šifrata C:

$$M = C^d \bmod n$$

Uslovi RSA algoritma

1. Moguće je naći vrednosti e, d, n takve da $M^{ed} \bmod n = M$ za svako $M < n$

Uslovi RSA algoritma

1. Moguće je naći vrednosti e, d, n takve da $M^{ed} \bmod n = M$ za svako $M < n$
2. $M^e \bmod n$ i $C^d \bmod n$ se lako računaju za sve vrednosti $M < n$

Uslovi RSA algoritma

1. Moguće je naći vrednosti e, d, n takve da $M^{ed} \bmod n = M$ za svako $M < n$
2. $M^e \bmod n$ i $C^d \bmod n$ se lako računaju za sve vrednosti $M < n$
3. Neizvodljivo odrediti d ako su dati e i n

Računska efikasnost RSA

- Šifrovanje i dešifrovanje zahtevaju računanje ekponencijalne funkcije

Računska efikasnost RSA

- Šifrovanje i dešifrovanje zahtevaju računanje ekponencijalne funkcije
  - Veoma veliki brojevi; koriste se karakteristike modularne aritmetike:

$$[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$$

Računska efikasnost RSA

- Šifrovanje i dešifrovanje zahtevaju računanje ekponencijalne funkcije
  - Veoma veliki brojevi; koriste se karakteristike modularne aritmetike:

$$[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$$

  - Brz i efikasan algoritam zasniva se na uzastopnom dizanju baze na kvadrat

Računska efikasnost RSA

- Šifrovanje i dešifrovanje zahtevaju računanje ekponencijalne funkcije
  - Veoma veliki brojevi; koriste se karakteristike modularne aritmetike:

$$[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$$

  - Brz i efikasan algoritam zasniva se na uzastopnom dizanju baze na kvadrat
- Biranje e

Računska efikasnost RSA

- Šifrovanje i dešifrovanje zahtevaju računanje ekponencijalne funkcije
  - Veoma veliki brojevi; koriste se karakteristike modularne aritmetike:

$$[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$$

  - Brz i efikasan algoritam zasniva se na uzastopnom dizanju baze na kvadrat
- Biranje e
  - Što manja vrednost to bolje performanse, ali i veća izčoženost napadima (popularna vrednost - 65537)

Računska efikasnost RSA

- Šifrovanje i dešifrovanje zahtevaju računanje ekponencijalne funkcije
  - Veoma veliki brojevi; koriste se karakteristike modularne aritmetike:

$$[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$$

  - Brz i efikasan algoritam zasniva se na uzastopnom dizanju baze na kvadrat
- Biranje e
  - Što manja vrednost to bolje performanse, ali i veća izčoženost napadima (popularna vrednost - 65537)
- Računanje d

Računska efikasnost RSA

- Šifrovanje i dešifrovanje zahtevaju računanje ekponencijalne funkcije
  - Veoma veliki brojevi; koriste se karakteristike modularne aritmetike:

$$[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$$

  - Brz i efikasan algoritam zasniva se na uzastopnom dizanju baze na kvadrat
- Biranje e
  - Što manja vrednost to bolje performanse, ali i veća izčoženost napadima (popularna vrednost - 65537)
- Računanje d
  - Malo d izloženo napadima

Računska efikasnost RSA

- Šifrovanje i dešifrovanje zahtevaju računanje ekponencijalne funkcije
  - Veoma veliki brojevi; koriste se karakteristike modularne aritmetike:

$$[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$$

  - Brz i efikasan algoritam zasniva se na uzastopnom dizanju baze na kvadrat
- Biranje e
  - Što manja vrednost to bolje performanse, ali i veća izčoženost napadima (popularna vrednost - 65537)
- Računanje d
  - Malo d izloženo napadima
  - Dosta brža dekripcija za veliko d primenom Kineske teoreme o ostacima i Fermaove teoreme

Sigurnost RSA

- Brute Force napad: izabrati veliko d

Sigurnost RSA

- Brute Force napad: izabrati veliko d
- Matematički napadi:

Sigurnost RSA

- Brute Force napad: izabrati veliko d
- Matematički napadi:
  1. Faktorisati n na njegova dva prosta činioca

Sigurnost RSA

- Brute Force napad: izabrati veliko d
- Matematički napadi:
  1. Faktorisati n na njegova dva prosta činioca
  2. Odrediti $\varphi(n)$ direktno, bez određivanja p ili q

Sigurnost RSA

- Brute Force napad: izabrati veliko d
- Matematički napadi:
  1. Faktorisati n na njegova dva prosta činioca
  2. Odrediti $\varphi(n)$ direktno, bez određivanja p ili q
  3. Odrediti d direktno, bez određivanja $\varphi(n)$

Sigurnost RSA

- Brute Force napad: izabrati veliko d
- Matematički napadi:
  1. Faktorisati n na njegova dva prosta činioca
  2. Odrediti $\varphi(n)$ direktno, bez određivanja p ili q
  3. Odrediti d direktno, bez određivanja $\varphi(n)$

  - Faktorisanje n se smatra najefikasnijim od ova tri pa se često koristi kao mera sigurnosti

Sigurnost RSA

- Brute Force napad: izabrati veliko d
- Matematički napadi:
  1. Faktorisati n na njegova dva prosta činioca
  2. Odrediti $\varphi(n)$ direktno, bez određivanja p ili q
  3. Odrediti d direktno, bez određivanja $\varphi(n)$

  - Faktorisanje n se smatra najefikasnijim od ova tri pa se često koristi kao mera sigurnosti
- Vremenski napadi: praktični, ali lako pobedivi (npr. konstantnim vremenom dekripcije)

Još neki kriptosistemi sa javnim ključem

ElGamal kriptosistem

- Sličan Diffie-Hellman-ovom

Kriptografija eliptičnih kriva

Još neki kriptosistemi sa javnim ključem

ElGamal kriptosistem

- Sličan Diffie-Hellman-ovom
- Koristi se u digitalnom potpisu

Kriptografija eliptičnih kriva

Još neki kriptosistemi sa javnim ključem

ElGamal kriptosistem

- Sličan Diffie-Hellman-ovom
- Koristi se u digitalnom potpisu

Kriptografija eliptičnih kriva

- Koristi aritmetiku eliptičnih kriva (umesto modularne u RSA)

Još neki kriptosistemi sa javnim ključem

ElGamal kriptosistem

- Sličan Diffie-Hellman-ovom
- Koristi se u digitalnom potpisu

Kriptografija eliptičnih kriva

- Koristi aritmetiku eliptičnih kriva (umesto modularne u RSA)
- Ista sigurnost kao RSA samo sa manjim ključevima

Još neki kriptosistemi sa javnim ključem

ElGamal kriptosistem

- Sličan Diffie-Hellman-ovom
- Koristi se u digitalnom potpisu

Kriptografija eliptičnih kriva

- Koristi aritmetiku eliptičnih kriva (umesto modularne u RSA)
- Ista sigurnost kao RSA samo sa manjim ključevima
- Koristi se za razmenu ključeva i digitalni potpis

Heš funkcija

- Heš funkcija - H: blok podataka varljive dužine M ulaz; heš vrednost konstantne dužine $h = \mathrm{H}(M)$ izlaz

Heš funkcija

- Heš funkcija - H: blok podataka varljive dužine M ulaz; heš vrednost konstantne dužine $h = \mathrm{H}(M)$ izlaz
- Kriptografska heš funkcija: računski neizvodljivo odrediti:

Heš funkcija

- Heš funkcija - H: blok podataka varljive dužine M ulaz; heš vrednost konstantne dužine $h = \mathrm{H}(M)$ izlaz
- Kriptografska heš funkcija: računski neizvodljivo odrediti:
  1. M koje se mapira u poznato h (karakteristika jednostranosti funkcije)

Heš funkcija

- Heš funkcija - H: blok podataka varljive dužine M ulaz; heš vrednost konstantne dužine $h = \mathrm{H}(M)$ izlaz
- Kriptografska heš funkcija: računski neizvodljivo odrediti:
  1. M koje se mapira u poznato h (karakteristika jednostranosti funkcije)
  2. M_1 i M_2 koji se mapiraju u isto h (bez kolizija)

Heš funkcija

- Heš funkcija - H: blok podataka varljive dužine M ulaz; heš vrednost konstantne dužine $h = \mathrm{H}(M)$ izlaz
- Kriptografska heš funkcija: računski neizvodljivo odrediti:
  1. M koje se mapira u poznato h (karakteristika jednostranosti funkcije)
  2. M_1 i M_2 koji se mapiraju u isto h (bez kolizija)
- Koristi se da bi odredili da li su se podaci promenili

Heš funkcija

- Heš funkcija - H: blok podataka varljive dužine M ulaz; heš vrednost konstantne dužine $h = \mathrm{H}(M)$ izlaz
- Kriptografska heš funkcija: računski neizvodljivo odrediti:
  1. M koje se mapira u poznato h (karakteristika jednostranosti funkcije)
  2. M_1 i M_2 koji se mapiraju u isto h (bez kolizija)
- Koristi se da bi odredili da li su se podaci promenili
- Primeri: autentifikacija poruke, digitalni potpis, detekcija virus-a...

Autentifikacija poruke

- Verifikuje integritet poruke

Autentifikacija poruke

- Verifikuje integritet poruke
  - Uverava da su primljeni podaci isti kao i poslati

Autentifikacija poruke

- Verifikuje integritet poruke
  - Uverava da su primljeni podaci isti kao i poslati
  - Obezbeđuje identitet pošiljaoca

Autentifikacija poruke

- Verifikuje integritet poruke
  - Uverava da su primljeni podaci isti kao i poslati
  - Obezbeđuje identitet pošiljaoca
- Postiže se heš funkcijom

O algoritmu

- 1995. NIST objavljuje SHA-1 nakon što su uočene slabosti algoritma SHA-0 objavljenog 1993.
- SHA-1 daje 160-bitnu heš vrednost
- izgrađen na istim principama kao i njegivi prethodnici MD4 i MD5
- iterativna procedura

Operacije

In the description of the hash algorithm, we need the following operations on strings of 32 bits:

1. $X \wedge Y$ = bitwise "and", which is bitwise multiplication mod 2, or bitwise minimum.
2. $X \vee Y$ = bitwise "or", which is bitwise maximum.
3. $X \oplus Y$ = bitwise addition mod 2.
4. $\neg X$ changes 1s to 0s and 0s to 1s .
5. $X + Y$ = addition of X and Y mod 2^{32}, where X and Y are regarded as integers mod 2^{32}.
6. $X \hookleftarrow r$ = shift of X to the left by r positions (and the beginning wraps around to the end).

We also need the following functions:

$$f_t(B,C,D) = \begin{cases} (B \wedge C) \vee ((\neg B) \wedge D) & \text{if } 0 \le t \le 19 \\ B \oplus C \oplus D & \text{if } 20 \le t \le 39 \\ (B \wedge C) \vee (B \wedge D) \vee (C \wedge D) & \text{if } 40 \le t \le 59 \\ B \oplus C \oplus D & \text{if } 60 \le t \le 79 \end{cases}$$

Define constants $K_0, \ldots, K_{79}$ as follows:

$$K_t = \begin{cases} \texttt{5A827999} & \text{if } 0 \le t \le 19 \\ \texttt{6ED9EBA1} & \text{if } 20 \le t \le 39 \\ \texttt{8F1BBCDC} & \text{if } 40 \le t \le 59 \\ \texttt{CA62C1D6} & \text{if } 60 \le t \le 79 \end{cases}$$

Algoritam

1. Poruka se deli u blokove od po 512 bita

Algoritam

1. Poruka se deli u blokove od po 512 bita
2. Inicijalizuju se vrednosti H_0,H_1,H_2,H_3,H_4

Algoritam

1. Poruka se deli u blokove od po 512 bita
2. Inicijalizuju se vrednosti H_0,H_1,H_2,H_3,H_4
3. Za svako i od 0 do $L-1$ gde je L broj blokova

Algoritam

1. Poruka se deli u blokove od po 512 bita
2. Inicijalizuju se vrednosti H_0,H_1,H_2,H_3,H_4
3. Za svako i od 0 do $L-1$ gde je L broj blokova
   - 3.1 Zapišimo $m_i = W_0||W_1||W_2||\ldots W_{14}||W_{15}$ gde svaki W_j ima 32 bita

Algoritam

1. Poruka se deli u blokove od po 512 bita
2. Inicijalizuju se vrednosti H_0,H_1,H_2,H_3,H_4
3. Za svako i od 0 do $L-1$ gde je L broj blokova
   - 3.1 Zapišimo $m_i = W_0||W_1||W_2||\ldots W_{14}||W_{15}$ gde svaki W_j ima 32 bita
   - 3.2 Za t od 16 do 79, neka je $W_t = (W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) \hookleftarrow 1$

Algoritam

1. Poruka se deli u blokove od po 512 bita
2. Inicijalizuju se vrednosti H_0,H_1,H_2,H_3,H_4
3. Za svako i od 0 do $L-1$ gde je L broj blokova
   - 3.1 Zapišimo $m_i = W_0||W_1||W_2||\ldots W_{14}||W_{15}$ gde svaki W_j ima 32 bita
   - 3.2 Za t od 16 do 79, neka je $W_t = (W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) \hookleftarrow 1$
   - 3.3 Neka je $A = H_0, B = H_1, C = H_2, D = H_3, E = H_4$

Algoritam

1. Poruka se deli u blokove od po 512 bita
2. Inicijalizuju se vrednosti H_0,H_1,H_2,H_3,H_4
3. Za svako i od 0 do $L-1$ gde je L broj blokova
   - 3.1 Zapišimo $m_i = W_0||W_1||W_2||\ldots W_{14}||W_{15}$ gde svaki W_j ima 32 bita
   - 3.2 Za t od 16 do 79, neka je $W_t = (W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) \hookleftarrow 1$
   - 3.3 Neka je $A = H_0, B = H_1, C = H_2, D = H_3, E = H_4$
   - 3.4 Za t od 0 do 79, $T = (A \hookleftarrow 5) + f_t(B,C,D) + E + W_t + K_t, E = D, D = C, C = (B \hookleftarrow 30), B = A, A = T$

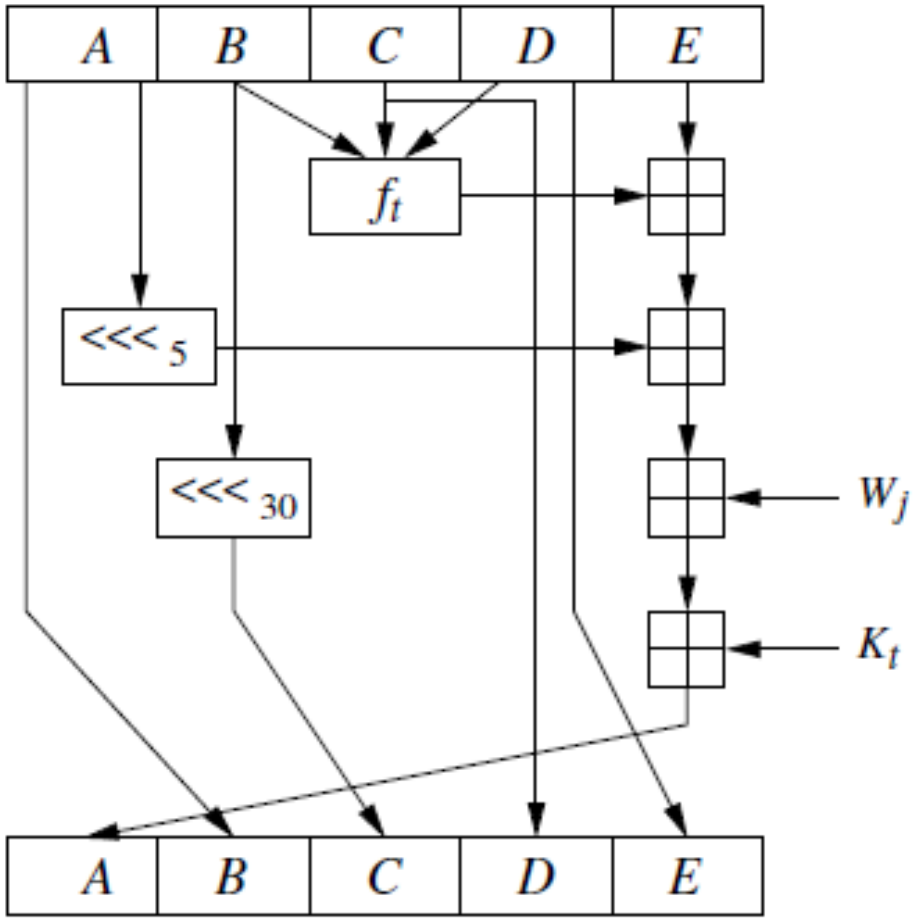
Algoritam

1. Poruka se deli u blokove od po 512 bita
2. Inicijalizuju se vrednosti H_0,H_1,H_2,H_3,H_4
3. Za svako i od 0 do $L-1$ gde je L broj blokova
   - 3.1 Zapišimo $m_i = W_0||W_1||W_2||\ldots W_{14}||W_{15}$ gde svaki W_j ima 32 bita
   - 3.2 Za t od 16 do 79, neka je $W_t = (W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) \hookleftarrow 1$
   - 3.3 Neka je $A = H_0, B = H_1, C = H_2, D = H_3, E = H_4$
   - 3.4 Za t od 0 do 79, $T = (A \hookleftarrow 5) + f_t(B,C,D) + E + W_t + K_t, E = D, D = C, C = (B \hookleftarrow 30), B = A, A = T$
   - 3.5 Neka je $H_0 = H_0 + A, H_1 = H_1 + B, H_2 = H_2 + C, H_3 = H_3 + D, H_4 = H_4 + E$

Algoritam

1. Poruka se deli u blokove od po 512 bita
2. Inicijalizuju se vrednosti H_0,H_1,H_2,H_3,H_4
3. Za svako i od 0 do $L-1$ gde je L broj blokova
   - 3.1 Zapišimo $m_i = W_0||W_1||W_2||\ldots W_{14}||W_{15}$ gde svaki W_j ima 32 bita
   - 3.2 Za t od 16 do 79, neka je $W_t = (W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) \hookleftarrow 1$
   - 3.3 Neka je $A = H_0, B = H_1, C = H_2, D = H_3, E = H_4$
   - 3.4 Za t od 0 do 79, $T = (A \hookleftarrow 5) + f_t(B,C,D) + E + W_t + K_t, E = D, D = C, C = (B \hookleftarrow 30)\,, B = A, A = T$
   - 3.5 Neka je $H_0 = H_0 + A, H_1 = H_1 + B, H_2 = H_2 + C, H_3 = H_3 + D, H_4 = H_4 + E$
4. Izlaz $H_0||H_1||H_2||H_3||H_4$

Algoritam



Još neke heš funkcije

- MD2
- MD4
- MD5
- SHA-0
- SHA-2 (skup heš funkcija)
- SHA-3 (trenutni standard (od 5. avgusta 2015. :)))

O digitalnom potpisu

- Cilj: pokazati bilo kome da je poruka potekla (ili je odobrena) od određenog korisnika

O digitalnom potpisu

- Cilj: pokazati bilo kome da je poruka potekla (ili je odobrena) od određenog korisnika
- Simetrična kriptografija

O digitalnom potpisu

- Cilj: pokazati bilo kome da je poruka potekla (ili je odobrena) od određenog korisnika
- Simetrična kriptografija
  - $Alisa$ i Bob dele tajni ključ K

O digitalnom potpisu

- Cilj: pokazati bilo kome da je poruka potekla (ili je odobrena) od određenog korisnika
- Simetrična kriptografija
  - $Alisa$ i Bob dele tajni ključ K
  - Primalac poruke (Alisa) može verifikovati da je poruka došla od drugog korisnika (Bob)

O digitalnom potpisu

- Cilj: pokazati bilo kome da je poruka potekla (ili je odobrena) od određenog korisnika
- Simetrična kriptografija
  - $Alisa$ i Bob dele tajni ključ K
  - Primalac poruke (Alisa) može verifikovati da je poruka došla od drugog korisnika (Bob)
  - Melet *ne može* znati da je poruka došla od Boba (mogla je doći i od Alise)

O digitalnom potpisu

- Cilj: pokazati bilo kome da je poruka potekla (ili je odobrena) od određenog korisnika
- Simetrična kriptografija
  - $Alisa$ i Bob dele tajni ključ K
  - Primalac poruke (Alisa) može verifikovati da je poruka došla od drugog korisnika (Bob)
  - Melet *ne može* znati da je poruka došla od Boba (mogla je doći i od Alise)
- Kriptografija sa javnim ključem može obezbediti potpis: samo jedan korisnik ima privatni ključ

Još neki algoritmi

- RSA
- ECDSA: DSA sa kriptografijom eliptičnih kriva
- ElGamal Signature Scheme: DSA je unapređenje ovog algoritma
- Različite heš funkcije se mogu koristiti