Algoritem RSA

Uporaba, prednosti in slabosti

Benjamin Benčina

Univerza v Ljubljani
Fakulteta za matematiko in fiziko
Oddelek za matematiko

24. maj 2018

Uvod v kriptografijo

- Umetnost skrivanja podatkov vsem na očeh.

Uvod v kriptografijo

- Umetnost skrivanja podatkov vsem na očeh.
- Tajne združbe, varnostne službe, vojska, dvorci, zločinci, intelektualna elita, znanstveniki, ugankarji, računalniški protokoli...

Uvod v kriptografijo

- Umetnost skrivanja podatkov vsem na očeh.
- Tajne združbe, varnostne službe, vojska, dvorci, zločinci, intelektualna elita, znanstveniki, ugankarji, računalniški protokoli...
- Kriptanaliza - matematična sestrična tradicionalne kriptografije

Kriptografija pred računalniki

- Tajne pisave, skitala, piktogrami, premetanke...

Kriptografija pred računalniki

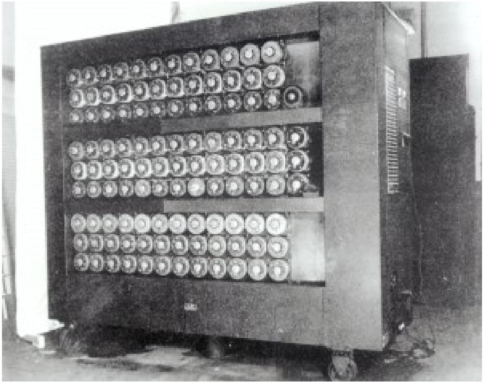
- Tajne pisave, skitala, piktogrami, premetanke...
- Cezarjanka ($x \mapsto x + c$)

Kriptografija pred računalniki

- Tajne pisave, skitala, piktogrami, premetanke...
- Cezarjanka ($x \mapsto x + c$)
- Vigenèrov kvadrat (urejena n-terica preslikav oblike $x \mapsto x + c_i$; kjer je n dolžina ključa in $i \in [n]$)

Kriptografija pred računalniki

- Tajne pisave, skitala, piktogrami, premetanke...
- Cezarjanka ($x \mapsto x + c$)
- Vigenèrov kvadrat (urejena n-terica preslikav oblike $x \mapsto x + c_i$; kjer je n dolžina ključa in $i \in [n]$)
- Enigma in Alan Turing



Slika: Ena od Turingovih bomb

Slika: Alan Turing, 16 let

Motivacija

- Ročne šifre so nepraktične, njihova varnost nezanesljiva.

Motivacija

- Ročne šifre so nepraktične, njihova varnost nezanesljiva.
- Mehanične šifre so drage s preveč dinamičnim algoritmom.

Motivacija

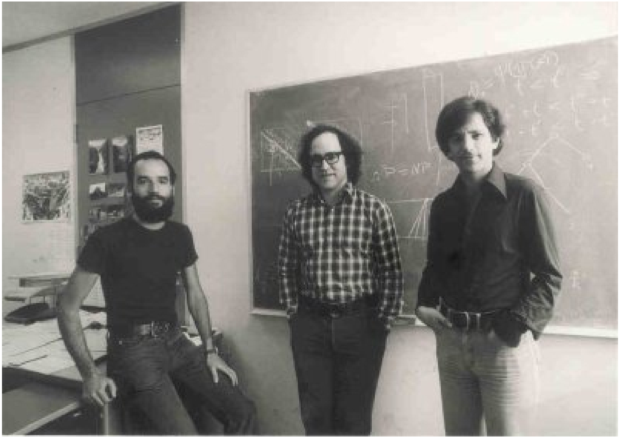
- Ročne šifre so nepraktične, njihova varnost nezanesljiva.
- Mehanične šifre so drage s preveč dinamičnim algoritmom.
- Obe vrsti tradicionalnega šifriranja sta proti računalniku skoraj vedno neuporabni.

Motivacija

- Ročne šifre so nepraktične, njihova varnost nezanesljiva.
- Mehanične šifre so drage s preveč dinamičnim algoritmom.
- Obe vrsti tradicionalnega šifriranja sta proti računalniku skoraj vedno neuporabni.

Potreba po univerzalnem, matematično trdnem in varnem algoritmu.

1978, IDEJA JE ROJENA



SLIKA: Izumitelji algoritma Ronald Rivest (sredina), Adi Shamir (levo) in Leonard Adleman (desno) po podelitvi patenta, 1983.

DEFINICIJA: **Modularna aritmetika** (včasih tudi urna aritmetika) po modulu n je aritmetika omejena s kongruenčno relacijo $aR_nb \iff n|b-a$.

Z drugimi besedami je to aritmetika v kolobarju $\mathbb{Z}_n$, tj. je kolobar ostankov pri deljenju celih števil z n, kjer je n **modul**.

Matematične osnove

Modularna aritmetika in kolobar $\mathbb{Z}_n$

DEFINICIJA: **Modularna aritmetika** (včasih tudi urna aritmetika) po modulu n je aritmetika omejena s kongruenčno relacijo $aR_nb \iff n|b-a$.

Z drugimi besedami je to aritmetika v kolobarju $\mathbb{Z}_n$, tj. je kolobar ostankov pri deljenju celih števil z n, kjer je n **modul**.

OZNAKE:

- Operacija mod: $\mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}$
  $(a, b) \mapsto$ ostanek števila a pri deljenju z b
- Relacija $a = b \pmod{n}$ pomeni, da celi števili a in b vrneta isti ostanek pri deljenju z n, oziroma $\mathrm{mod}(a,n) = \mathrm{mod}(b,n)$.

Matematične osnove

Funkcija φ in Eulerjev izrek

DEFINICIJA: Eulerjeva funkcija $\varphi(n)$ vrne število vseh pozitivnih celih števil manjših od n, ki so n tuja.

$\varphi(n) = \#\{\ a \in \mathbb{N};\ a \leq n,\ \gcd(a,\ n) = 1\}$

DEFINICIJA: Eulerjeva funkcija $\varphi(n)$ vrne število vseh pozitivnih celih števil manjših od n, ki so n tuja.

$\varphi(n) = \#\{ a \in \mathbb{N}; a \leq n, \gcd(a, n) = 1\}$

EULERJEV IZREK: Če sta si števili x in n tuji, velja:

$$x^{\varphi(n)} = 1 \pmod n$$

Matematične osnove

Funkcija φ in Eulerjev izrek

DEFINICIJA: Eulerjeva funkcija $\varphi(n)$ vrne število vseh pozitivnih celih števil manjših od n, ki so n tuja.

$\varphi(n) = \#\{\ a \in \mathbb{N};\ a \leq n,\ \gcd(a,\ n) = 1\}$

EULERJEV IZREK: Če sta si števili x in n tuji, velja:

$$x^{\varphi(n)} = 1 \pmod{n}$$

OPOMBA: $\varphi(p) = p - 1$, če je p praštevilo.

Matematične osnove

Polinomi in številski sistemi

DEFINICIJA: Polinom f je formalna vsota $f(X) = a_0 + a_1 X + \cdots + a_n X^n$. X imenujemo *spremenljivka*, številom a_i pravimo *koeficienti*, n pa je *stopnja* polinoma. Vsak polinom porodi **polinomsko funkcijo**.

DEFINICIJA: Polinom f je formalna vsota $f(X) = a_0 + a_1X + \cdots + a_nX^n$. X imenujemo *spremenljivka*, številom a_i pravimo *koeficienti*, n pa je *stopnja* polinoma. Vsak polinom porodi **polinomsko funkcijo**.

DEFINICIJA: Število je zapisano v n-iškem **številskem sistemu**, $n \geq 2$, če je enako vrednosti neke polinomske funkcije iz $\mathbb{Z}_n$ izračunane za $X = n$ in je a_i števka tega števila na i-tem mestu, šteto od desne proti levi od 0 naprej.

Kako deluje?

- Izberemo si dve različni praštevili p in q.

Kako deluje?

- Izberemo si dve različni praštevili p in q.
- Izračunamo $n = p \cdot q$ in $\phi(n) = \phi(p) \cdot \phi(q) = (p-1) \cdot (q-1)$.

Kako deluje?

- Izberemo si dve različni praštevili p in q.
- Izračunamo $n = p \cdot q$ in $\phi(n) = \phi(p) \cdot \phi(q) = (p-1) \cdot (q-1)$.
- Izberemo naključen e, za katerega velja $\gcd(e, \phi(n)) = 1$.

Kako deluje?

- Izberemo si dve različni praštevili p in q.
- Izračunamo $n = p \cdot q$ in $\phi(n) = \phi(p) \cdot \phi(q) = (p-1) \cdot (q-1)$.
- Izberemo naključen e, za katerega velja $\gcd(e, \phi(n)) = 1$.
- Z razširjenim Evklidovim algoritmom poiščemo d, ki je multiplikativen inverz za e v kolobarju $\mathbb{Z}_{\phi(n)}$. Drugače: $e \cdot d = 1 \pmod{\phi(n)}$.

Kako deluje?

- Izberemo si dve različni praštevili p in q.
- Izračunamo $n = p \cdot q$ in $\phi(n) = \phi(p) \cdot \phi(q) = (p-1) \cdot (q-1)$.
- Izberemo naključen e, za katerega velja $\gcd(e, \phi(n)) = 1$.
- Z razširjenim Evklidovim algoritmom poiščemo d, ki je multiplikativen inverz za e v kolobarju $\mathbb{Z}_{\phi(n)}$. Drugače: $e \cdot d = 1 \pmod{\phi(n)}$.
- Sporočilo m šifriramo tako: $c = m^e \bmod n$.

Kako deluje?

- Izberemo si dve različni praštevili p in q.
- Izračunamo $n = p \cdot q$ in $\phi(n) = \phi(p) \cdot \phi(q) = (p-1) \cdot (q-1)$.
- Izberemo naključen e, za katerega velja $\gcd(e, \phi(n)) = 1$.
- Z razširjenim Evklidovim algoritmom poiščemo d, ki je multiplikativen inverz za e v kolobarju $\mathbb{Z}_{\phi(n)}$. Drugače: $e \cdot d = 1 \pmod{\phi(n)}$.
- Sporočilo m šifriramo tako: $c = m^e \bmod n$.
- Skrito sporočilo dešifriramo tako: $m = c^d \bmod n$.

Zakaj deluje?

IZREK: Naj bo $n \in \mathbb{Z}$ produkt samih različnih praštevil in naj bosta $d, e \in \mathbb{N}$ taki, da za $\forall p \in \mathbb{P}$, kjer $p|n$, velja: $(p-1)|(d \cdot e - 1)$. Tedaj:

$$\forall a \in \mathbb{Z}: \ \ a^{d \cdot e} = a \pmod n$$

Kodiranje zaporedja znakov v števila

- Izračunamo moč množice znakov, ki bi jih radi kodirali. Naj bo to število m.

Kodiranje zaporedja znakov v števila

- Izračunamo moč množice znakov, ki bi jih radi kodirali. Naj bo to število m.
- Vsakemu znaku iz te množice priredimo vrednost iz množice $\mathbb{Z}_m$.

Kodiranje zaporedja znakov v števila

- Izračunamo moč množice znakov, ki bi jih radi kodirali. Naj bo to število m.
- Vsakemu znaku iz te množice priredimo vrednost iz množice $\mathbb{Z}_m$.
- Preštejemo število znakov, ki bi jih radi kodirali. Naj bo to število j.

Kodiranje zaporedja znakov v števila

- Izračunamo moč množice znakov, ki bi jih radi kodirali. Naj bo to število m.
- Vsakemu znaku iz te množice priredimo vrednost iz množice $\mathbb{Z}_m$.
- Preštejemo število znakov, ki bi jih radi kodirali. Naj bo to število j.
- Zaporedje znakov razbijemo v take kose, da je j vsakega kosa strogo manj od $\log_m(n)$.
- Nato sestavimo polinom $f(X) = a_0 + a_1 \cdot X + \cdots + a_j \cdot X^j$.

Kodiranje zaporedja znakov v števila

- Izračunamo moč množice znakov, ki bi jih radi kodirali. Naj bo to število m.
- Vsakemu znaku iz te množice priredimo vrednost iz množice $\mathbb{Z}_m$.
- Preštejemo število znakov, ki bi jih radi kodirali. Naj bo to število j.
- Zaporedje znakov razbijemo v take kose, da je j vsakega kosa strogo manj od $\log_m(n)$.
- Nato sestavimo polinom $f(X) = a_0 + a_1 \cdot X + \cdots + a_j \cdot X^j$.
- Naše iskano število je vrednost polinomske funkcije tega polinoma izračunane za $x = m$.

Načrt napada

Iz matematičnega vidika se da algoritem napasti na treh točkah:

- določimo praštevili p in q modula n, nato izračunamo $\phi(n)$ in posledično d;

Načrt napada

Iz matematičnega vidika se da algoritem napasti na treh točkah:

- določimo praštevili p in q modula n, nato izračunamo $\phi(n)$ in posledično d;
- določimo $\phi(n)$ direktno iz javnega ključa $\{e, n\}$ in posledično d;

Načrt napada

Iz matematičnega vidika se da algoritem napasti na treh točkah:

- določimo praštevili p in q modula n, nato izračunamo $\phi(n)$ in posledično d;
- določimo $\phi(n)$ direktno iz javnega ključa $\{e, n\}$ in posledično d;
- določimo d direktno iz javnega ključa $\{e, n\}$.

Načrt napada

Iz matematičnega vidika se da algoritem napasti na treh točkah:

- določimo praštevili p in q modula n, nato izračunamo $\phi(n)$ in posledično d;
- določimo $\phi(n)$ direktno iz javnega ključa $\{e, n\}$ in posledično d;
- določimo d direktno iz javnega ključa $\{e, n\}$.

Pokazati se da, da je zahtevnost drugega in tretjega pristopa enaka zahtevnosti faktorizacije števila n. Torej je varnost algoritma proti matematičnim napadom določena s časom potrebnim za faktorizacijo števila n na p in q.

Napad 1:

Faktorizacija n, če poznamo $\phi(n)$

- Naj bo $n = p \cdot q$ in $\phi(n)$ znano število.

Napad 1:

Faktorizacija n, če poznamo $\phi(n)$

- Naj bo $n = p \cdot q$ in $\phi(n)$ znano število.
- Velja $\phi(n) = (p-1)\cdot(q-1) = p \cdot q - (p+q) + 1$, torej poznamo tako $p \cdot q = n$ kot $p + q = n + 1 - \phi(n)$.

Napad 1:

Faktorizacija n, če poznamo $\phi(n)$

- Naj bo $n = p \cdot q$ in $\phi(n)$ znano število.
- Velja $\phi(n) = (p-1) \cdot (q-1) = p \cdot q - (p+q) + 1$, torej poznamo tako $p \cdot q = n$ kot $p + q = n + 1 - \phi(n)$.
- Torej po Vietovih formulah vemo, da velja $x^2 - (p+q) \cdot x + p \cdot q = (x-p) \cdot (x-q)$.

Napad 1:

Faktorizacija n, če poznamo $\phi(n)$

- Naj bo $n = p \cdot q$ in $\phi(n)$ znano število.
- Velja $\phi(n) = (p-1)\cdot(q-1) = p\cdot q - (p+q) + 1$, torej poznamo tako $p \cdot q = n$ kot $p + q = n + 1 - \phi(n)$.
- Torej po Vietovih formulah vemo, da velja $x^2 - (p+q)\cdot x + p\cdot q = (x-p)\cdot(x-q)$.
- p in q sta očitno ničli te kvadratne funkcije in zato lahko izračunljivi po splošni formuli.

Napad 2:

Kaj če sta p in q blizu skupaj?

- Denimo, da vemo, da je razlika števil p in q majhna. Potem je n lahko faktorizirati s *Fermatovo faktorizacijsko metodo*.

Napad 2:

Kaj če sta p in q blizu skupaj?

- Denimo, da vemo, da je razlika števil p in q majhna. Potem je n lahko faktorizirati s *Fermatovo faktorizacijsko metodo*.
- Brez izgube splošnosti lahko rečemo, da $p > q$. Potem $n = (\frac{p+q}{2})^2 - (\frac{p-q}{2})^2$.

Napad 2:

Kaj če sta p in q blizu skupaj?

- Denimo, da vemo, da je razlika števil p in q majhna. Potem je n lahko faktorizirati s *Fermatovo faktorizacijsko metodo*.
- Brez izgube splošnosti lahko rečemo, da $p > q$. Potem $n = (\frac{p+q}{2})^2 - (\frac{p-q}{2})^2$.
- Ker sta p in q "blizu", je $s = \frac{p-q}{2}$ majhno število, $t = \frac{p+q}{2}$ pa je le malce večje od $\sqrt{n}$, $t^2 - n = s^2$ pa je popoln kvadrat.

Napad 2:

Kaj če sta p in q blizu skupaj?

- Denimo, da vemo, da je razlika števil p in q majhna. Potem je n lahko faktorizirati s *Fermatovo faktorizacijsko metodo*.
- Brez izgube splošnosti lahko rečemo, da $p > q$. Potem $n = (\frac{p+q}{2})^2 - (\frac{p-q}{2})^2$.
- Ker sta p in q "blizu", je $s = \frac{p-q}{2}$ majhno število, $t = \frac{p+q}{2}$ pa je le malce večje od $\sqrt{n}$, $t^2 - n = s^2$ pa je popoln kvadrat.
- Za t jemljemo $\lceil \sqrt{n} \rceil + k$, kjer $k \in \{0, 1, \dots\}$, dokler $t^2 - n$ ni popoln kvadrat.

Napad 2:

Kaj če sta p in q blizu skupaj?

- Denimo, da vemo, da je razlika števil p in q majhna. Potem je n lahko faktorizirati s *Fermatovo faktorizacijsko metodo*.
- Brez izgube splošnosti lahko rečemo, da $p > q$. Potem $n = (\frac{p+q}{2})^2 - (\frac{p-q}{2})^2$.
- Ker sta p in q "blizu", je $s = \frac{p-q}{2}$ majhno število, $t = \frac{p+q}{2}$ pa je le malce večje od $\sqrt{n}$, $t^2 - n = s^2$ pa je popoln kvadrat.
- Za t jemljemo $\lceil \sqrt{n} \rceil + k$, kjer $k \in \{0, 1, \dots\}$, dokler $t^2 - n$ ni popoln kvadrat.
- Potem velja: $p = t + s$ in $q = t - s$.

Napad 3:

Faktorizacija n, če poznamo d, oz. napad na skupen modul

Pokažimo, da je iskanje d vsaj tako težko kot faktorizacija n. Recimo, da d imamo. Naj bo $m = d \cdot e - 1$. Torej velja $a^m = 1 \pmod{n}$.

Napad 3:

Faktorizacija n, če poznamo d, oz. napad na skupen modul

Pokažimo, da je iskanje d vsaj tako težko kot faktorizacija n. Recimo, da d imamo. Naj bo $m = d \cdot e - 1$. Torej velja $a^m = 1 \pmod{n}$.

ALGORITEM:

- Če je m sod in $a^{m/2} = 1 \pmod{n}$ za več naključnih a-jev, nastavimo $m = \frac{m}{2}$. Ponavljamo dokler so pogoji na m zadoščeni.

Napad 3:

Faktorizacija n, če poznamo d, oz. napad na skupen modul

Pokažimo, da je iskanje d vsaj tako težko kot faktorizacija n. Recimo, da d imamo. Naj bo $m = d \cdot e - 1$. Torej velja $a^m = 1 \pmod{n}$.

ALGORITEM:

- Če je m sod in $a^{m/2} = 1 \pmod{n}$ za več naključnih a-jev, nastavimo $m = \frac{m}{2}$. Ponavljamo dokler so pogoji na m zadoščeni.
- Izberemo naključen a in izračunamo $g = \gcd(a^{m/2} - 1, n)$.

Napad 3:

Faktorizacija n, če poznamo d, oz. napad na skupen modul

Pokažimo, da je iskanje d vsaj tako težko kot faktorizacija n. Recimo, da d imamo. Naj bo $m = d \cdot e - 1$. Torej velja $a^m = 1 \pmod{n}$.

ALGORITEM:

- Če je m sod in $a^{m/2} = 1 \pmod{n}$ za več naključnih a-jev, nastavimo $m = \frac{m}{2}$. Ponavljamo dokler so pogoji na m zadoščeni.
- Izberemo naključen a in izračunamo $g = \gcd(a^{m/2} - 1, n)$.
- Če je g pravi delitelj n, smo našli vrednost in zaključimo. Drugače nazaj na drugi korak.

Napad 3:

Faktorizacija n, če poznamo d, oz. napad na skupen modul

Pokažimo, da je iskanje d vsaj tako težko kot faktorizacija n. Recimo, da d imamo. Naj bo $m = d \cdot e - 1$. Torej velja $a^m = 1 \pmod{n}$.

ALGORITEM:

- Če je m sod in $a^{m/2} = 1 \pmod{n}$ za več naključnih a-jev, nastavimo $m = \frac{m}{2}$. Ponavljamo dokler so pogoji na m zadoščeni.
- Izberemo naključen a in izračunamo $g = \gcd(a^{m/2} - 1, n)$.
- Če je g pravi delitelj n, smo našli vrednost in zaključimo. Drugače nazaj na drugi korak.

Dve osebi naj zato nikoli ne uporabljata istega n.

Napad 4:
Chosen-Ciphertext Attack

- Preprost napad, ki temelji na standardni velikosti blokov sporočila.

Napad 4:
Chosen-Ciphertext Attack

- Preprost napad, ki temelji na standardni velikosti blokov sporočila.
- Napadalec določi množico možnih sporočil, izbere najbolj verjetne, jih zakodira z javnim ključem in nato primerja šifro (oz. po ang. Ciphertext).

Napad 4:
Chosen-Ciphertext Attack

- Preprost napad, ki temelji na standardni velikosti blokov sporočila.
- Napadalec določi množico možnih sporočil, izbere najbolj verjetne, jih zakodira z javnim ključem in nato primerja šifro (oz. po ang. Ciphertext).
- Rešitve:

Napad 4:
Chosen-Ciphertext Attack

- Preprost napad, ki temelji na standardni velikosti blokov sporočila.
- Napadalec določi množico možnih sporočil, izbere najbolj verjetne, jih zakodira z javnim ključem in nato primerja šifro (oz. po ang. Ciphertext).
- Rešitve:
  - Sporočilo *posolimo*, tj. mu na začetek vsakega bloka dodamo dogovorjeno število naključnih znakov.

Napad 4:
Chosen-Ciphertext Attack

- Preprost napad, ki temelji na standardni velikosti blokov sporočila.
- Napadalec določi množico možnih sporočil, izbere najbolj verjetne, jih zakodira z javnim ključem in nato primerja šifro (oz. po ang. Ciphertext).
- Rešitve:
  - Sporočilo *posolimo*, tj. mu na začetek vsakega bloka dodamo dogovorjeno število naključnih znakov.
  - Poskrbimo, da je naša izbira javnega ključa takšna, da je množica možnih sporočil kar se da velika.

Napad 5:

Napad na majhen šifrirni eksponent e

Mogoče se zdi, da je zaradi računske učinkovitosti pametno izbrati čim manjši e.

Napad 5:
Napad na majhen šifrirni eksponent e

Mogoče se zdi, da je zaradi računske učinkovitosti pametno izbrati čim manjši e.

NAPAKA!

Napad 5:

Napad na majhen šifrirni eksponent e

Mogoče se zdi, da je zaradi računske učinkovitosti pametno izbrati čim manjši e.

NAPAKA!

- Če ima tudi sporočilo m majhno vrednost, torej da $m^e < n$, potem tudi $c = m^e$. Torej lahko iz šifre c dobimo izvorno sporočilo zlahka tako: $m = c^{1/e}$.

Napad 5:

Napad na majhen šifrirni eksponent e

Mogoče se zdi, da je zaradi računske učinkovitosti pametno izbrati čim manjši e.

NAPAKA!

- Če ima tudi sporočilo m majhno vrednost, torej da $m^e < n$, potem tudi $c = m^e$. Torej lahko iz šifre c dobimo izvorno sporočilo zlahka tako: $m = c^{1/e}$.

Rešitev: premajhno sporočilo spet lahko *posolimo*, da povečamo $m^e > n$, poleg tega pa je vedno pametno izbrati večji e.

Napad 6:

Napad na majhen prostor sporočil

- Če je slika preslikave šifriranja pri izbranem javnem ključu premajhna, lahko napadalec sistem očitno učinkovito napade že s surovo silo (*brute force*), saj lahko preprosto izračuna vse šifrirane nize in jih primerja s prestreženim.

Napad 6:

Napad na majhen prostor sporočil

- Če je slika preslikave šifriranja pri izbranem javnem ključu premajhna, lahko napadalec sistem očitno učinkovito napade že s surovo silo (*brute force*), saj lahko preprosto izračuna vse šifrirane nize in jih primerja s prestreženim.
- Na velikost prostora sporočil lahko vpliva veliko parametrov, predvsem izbira praštevil p in q, izbira e ter posledično "izbira"d.

Napad 6:

Napad na majhen prostor sporočil

- Če je slika preslikave šifriranja pri izbranem javnem ključu premajhna, lahko napadalec sistem očitno učinkovito napade že s surovo silo (*brute force*), saj lahko preprosto izračuna vse šifrirane nize in jih primerja s prestreženim.
- Na velikost prostora sporočil lahko vpliva veliko parametrov, predvsem izbira praštevil p in q, izbira e ter posledično "izbira"d.
- Ta napad je zato kombinacija tehnik Chosen-Ciphertext napada, napada na majhen šifrirni eksponent in naslednjega napada.

Napad 6:

Napad na majhen prostor sporočil

- Če je slika preslikave šifriranja pri izbranem javnem ključu premajhna, lahko napadalec sistem očitno učinkovito napade že s surovo silo (*brute force*), saj lahko preprosto izračuna vse šifrirane nize in jih primerja s prestreženim.
- Na velikost prostora sporočil lahko vpliva veliko parametrov, predvsem izbira praštevil p in q, izbira e ter posledično "izbira"d.
- Ta napad je zato kombinacija tehnik Chosen-Ciphertext napada, napada na majhen šifrirni eksponent in naslednjega napada.

Splošna rešitev je spet naključno *soljenje* sporočila in predvsem pametna izbira ključev.

Napad 7:

Napad na majhen dešifrirni eksponent d

Zopet bi se mogoče zdelo smiselno izbrati čim manjši d v imenu računske učinkovitosti.

Napad 7:

Napad na majhen dešifrirni eksponent d

Zopet bi se mogoče zdelo smiselno izbrati čim manjši d v imenu računske učinkovitosti.

NAPAKA!

Napad 7:

Napad na majhen dešifrirni eksponent d

Zopet bi se mogoče zdelo smiselno izbrati čim manjši d v imenu računske učinkovitosti.

NAPAKA!

Kriptograf Michael J. Wiener je pokazal, da se da d da učinkovito določiti, če $d < \frac{1}{3} \cdot n^{1/4}$ in če za praštevili p in q velja $q < p < 2 \cdot q$.

Napad 7:

Napad na majhen dešifrirni eksponent d

Zopet bi se mogoče zdelo smiselno izbrati čim manjši d v imenu računske učinkovitosti.

NAPAKA!

Kriptograf Michael J. Wiener je pokazal, da se da d da učinkovito določiti, če $d < \frac{1}{3} \cdot n^{1/4}$ in če za praštevili p in q velja $q < p < 2 \cdot q$.

Rešitev je očitna: eksponent e moramo pazljivo izbrati tako, da $d > \frac{1}{3} \cdot n^{1/4}$

Napad 8:
Cikličen napad

CIkličen napad je idejno najpreprostejši napad na algoritem RSA poleg seveda *brute force* napada:

- Šifro z javnim ključem še enkrat šifriramo. Ta korak ponavljamo, dokler ne šifriramo dobljenega števila nazaj v izvorno šifro.

Napad 8:
Cikličen napad

CIkličen napad je idejno najpreprostejši napad na algoritem RSA poleg seveda *brute force* napada:

- Šifro z javnim ključem še enkrat šifriramo. Ta korak ponavljamo, dokler ne šifriramo dobljenega števila nazaj v izvorno šifro.
- Ko se to zgodi, pogledamo eno šifriranje nazaj, to število mora biti izvirno sporočilo.

Napad 8:
Cikličen napad

CIkličen napad je idejno najpreprostejši napad na algoritem RSA poleg seveda *brute force* napada:

- Šifro z javnim ključem še enkrat šifriramo. Ta korak ponavljamo, dokler ne šifriramo dobljenega števila nazaj v izvorno šifro.
- Ko se to zgodi, pogledamo eno šifriranje nazaj, to število mora biti izvirno sporočilo.

Ta napad je ob pravilni izbiri javnega ključa časovno ekvivalenten *brute force* napadu.
Njegova edina dobra lastnost je, da je odporen na *soljenje* sporočila,

Uporaba

- OpenSSH protokol.

Uporaba

- OpenSSH protokol.
- Če obrnemo vlogo prejemnika in pošiljatelja, lahko algoritem RSA nadomešča t.i. kriptografsko *hash funkcijo* pri preverjanju istovetnosti.

Uporaba

- OpenSSH protokol.
- Če obrnemo vlogo prejemnika in pošiljatelja, lahko algoritem RSA nadomešča t.i. kriptografsko *hash funkcijo* pri preverjanju istovetnosti.
- Zaradi visoke računske zahtevnosti je algoritem pogosto prepočasen za pošiljanje podatkov v realnem času, zato pa pogosto igra vlogo anonimizatorja predaje ključa za hitrejše simetrične šifre, namesto recimo *Diffie-Hellmanove izmenjave ključa*.

HVALA ZA POZORNOST!