Turinys

Minimalus-HTTPS

(angl. *Mini-Hyper-text transfer protocol secure*, *Mini-Https*)

Minimalus (HTTPS principo supratimui supaprastintas paaiškinimas)-HTTPS yra HTTP protokolo saugi versija, įtraukianti SSL/TLS (angl. *Secure Socket Layer/Transport Layer Security*) šifravimą. Šis šifravimas užtikrina, kad visi duomenys, siunčiami tarp naršyklės ir serverio, yra užšifruoti ir saugūs nuo perėmimo ar modifikavimo. Be to, užtikrina informacijos konfidencialumą, vientisumą ir autentiškumą, net ir naudojant atvirus ryšio kanalus.

HTTPS ypač svarbus svetainėse, kuriose apdorojama asmeninė ar finansinė informacija, pavyzdžiui, internetinėse parduotuvėse ar bankų svetainėse.

Viešųjų parametrų apskaičiavimas

Viešieji parametrai **PP**=(***p***, ***g***)

Apskritai sudėtinga užduotis rasti generatorius aibėje $\mathbf{Z}_p^* = \{1, 2, 3, \ldots, p-1\}$, tačiau naudojant stiprų pirminį ***p*** ir *Lagranžo teoremą grupės teorijoje*, generatorių $\mathbf{Z}_p^*$ galima rasti atsitiktine tvarka. Paieška laikoma užbaiga jei tenkinamos dvi sąlygos:

1. jeigu ***p*** ir ***q*** yra stiprūs pirminiai $p = 2 \cdot q + 1 \rightarrow q = (p-1)/2$;
2. jeigu visi $g \in \Gamma$, $g^q \neq 1 \ mod\ p$ ir $g^2 \neq 1 \ mod\ p$. **Tik 40% skaičių yra generatoriai.**

Pavyzdinis generatoriaus radimas (***g*** didinamas po vienetą, kol *ans* nelygus 1 ir neviršija ***p***):

```
>> p=genstrongprime(28)
p = 187086587
>> isprime(p)
ans = 1
>> q=(p-1)/2
q = 93543293
>> isprime(q)
ans = 1
>> g=2
>> mod_exp(g,q,p)
ans = 187086586
>> g=3;
>> mod_exp(g,q,p)
ans = 1
>> g=4;
>> mod_exp(g,q,p)
ans = 1
```

```
>> p=genstrongprime(28)
p = 144668519
>> q=(p-1)/2
q = 72334259
>> g=2;
>> mod_exp(g,q,p)
ans = 1
>> g=7;
>> mod_exp(g,q,p)
ans = 144668518
```

```
>> p=genstrongprime(28)
p = 224013599
>> q=(p-1)/2
q = 112006799
>> g=111;
>> mod_exp(g,q,p)
ans = 224013598
```

Toliau naudosime ***p***=int64(144668519); ***g***=7.

Minimalus-HTTPS

Aldonos kreipimąsi į **Banką** vaizduojanti schema pateikiama 1 pav.

1.1 Pasirinkti atsitiktinį privatų raktą

$$PR_A = x,\ x \leftarrow \text{randi}(Z_{p-1})$$

ir apskaičiuoti viešą raktą:

$$VR_A = a = g^x \bmod p$$

1.2 Pasirinkti atsitiktinį skaičių

$$u \leftarrow \text{randi}(Z_{p-1})$$

ir apskaičiuoti sesijos viešą parametrą:

$$t_A = g^u \bmod p;$$

1.3 Pasirašyti t_A su Šnoro parašu

$\sigma = (r_A, s_A)$: pasirinkti atsitiktinį skaičių

$$i \leftarrow \text{randi}(Z_{p-1})$$

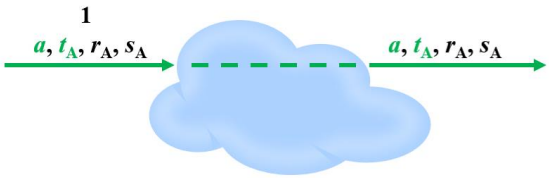
1.4 Apskaičiuoti pirmą parašo komponentę:

$$r_A = g^i \bmod p;$$

1.5 Apskaičiuoti antrą parašo komponentę:

$$h = \text{hash}(\text{concat}(t_A, r_A)),$$

$$s_A = (i + xh) \bmod (p - 1).$$



1 pav. Aldona kreipiasi į Banką

Aldonos veiksmai kreipiantis į **Banką**:

1.1. Pasirinkti atsitiktinį privatų raktą $PR_A = x,\ x \leftarrow \text{randi}(Z_{p-1})$ ir apskaičiuoti viešą raktą: $VR_A = a = g^x \bmod p$:

```
>> x=int64(randi(p-1))                    >> a=mod_exp(g,x,p)
x = 93976157                              a = 139483450
```

1.2. Pasirinkti atsitiktinį skaičių $u \leftarrow \text{randi}(Z_{p-1})$ ir apskaičiuoti sesijos viešą parametrą $t_A = g^u \bmod p$:

```
>> u=int64(randi(p-1))                    >> tA=mod_exp(g,u,p)
u = 74837953                              tA = 48755790
```

1.3. Pasirašyti t_A su Šnoro parašu $\sigma_A = (r_A, s_A)$ pasirinkus atsitiktinį skaičių $i \leftarrow \text{randi}Z(_{p-1})$, kad $1 < i < p-1$:

1.3.1. Apskaičiuoti pirmą parašo komponentę: $r_A = g^i \bmod p$:

```
>> i=int64(randi(p-1))                    >> rA=mod_exp(g,i,p)
i = 142081823                             rA = 140311641
>> 1 < i & i < p-1
ans = 1
```

1.3.2. Sujungti (angl. concat) t_A ir r_A, apskaičiuojant santrauką $h_A = H(t_A || r_A)$ ir apskaičiuoti antrąją parašo komponentę $s_A = (i + xh) \bmod (p - 1)$:

```
>> hA = hd28(concat(tA, rA))              >> sA=mod((i+x*hA),p-1)
hA = 96291913                             sA = 98793216
```

1.3.3. Parašas h_A santraukai yra $\sigma_A = (r_A, s_A)$

Pasirašyti$(x, h_A) = \sigma_A = (r_A, s_A) = (140311641, 98793216)$.

1.4. Aldona siunčia **Bankui** savo viešą raktą b, sesijos viešą parametrą t_B ir šiam parametrui suformuotą parašą σ_A.

Banko atsakymą į **Aldonos** kreipimąsi vaizduojanti schema pateikiama 2 pav.

1.1 Pasirinkti atsitiktinį privatų raktą
$PR_A = x$, $x \leftarrow randi(Z_{p-1})$
ir apskaičiuoti viešą raktą:
$VR_A = a = g^x \bmod p$
1.2 Pasirinkti atsitiktinį skaičių
$u \leftarrow randi(Z_{p-1})$
ir apskaičiuoti sesijos viešą parametrą:
$t_A = g^u \bmod p$;
1.3 Pasirašyti t_A su Šnoro parašu
$\sigma = (r_A, s_A)$: pasirinkti atsitiktinį skaičių
$i \leftarrow randi(Z_{p-1})$
1.4 Apskaičiuoti pirmą parašo komponentę:
$r_A = g^i \bmod p$;
1.5 Apskaičiuoti antrą parašo komponentę:
$h = hash(concat(t_A, r_A))$,
$s_A = (i + xh) \bmod (p - 1)$.

1
a, t_A, r_A, s_A →

→ a, t_A, r_A, s_A

2.1 Pasirinkti atsitiktinį privatų raktą
$PR_B = y$, $y \leftarrow randi(Z_{p-1})$
ir apskaičiuoti viešą raktą:
$VR_B = b = g^y \bmod p$
2.2 Pasirinkti atsitiktinį skaičių
$v \leftarrow randi(Z_{p-1})$
ir apskaičiuoti sesijos viešą parametrą:
$t_B = g^v \bmod p$;
2.3 Pasirašyti t_B su Šnoro parašu
$\sigma = (r_B, s_B)$: pasirinkti atsitiktinį skaičių
$z \leftarrow randi(Z_{p-1})$
2.4 Apskaičiuoti pirmą parašo komponentę:
$r_B = g^z \bmod p$;
2.5 Apskaičiuoti antrą parašo komponentę:
$h = hash(concat(t_B, r_B))$,
$s_B = (z + yh) \bmod (p - 1)$.

2 pav. Bankas atsako Aldonai

Banko veiksmai atsakant **Aldonai**:

2.2. Pasirinkti atsitiktinį privatų raktą $PR_B = y$, $y \leftarrow randi(Z_{p-1})$ ir apskaičiuoti viešą raktą: $VR_B = b = g^y \bmod p$:

```
>> y=int64(randi(p-1))          >> b=mod_exp(g,y,p)
y = 122477781                   b = 88951568
```

2.3. Pasirinkti atsitiktinį skaičių $v \leftarrow randi(Z_{p-1})$ ir apskaičiuoti sesijos viešą parametrą $t_B = g^v \bmod p$:

```
>> v=int64(randi(p-1))          >> tB=mod_exp(g,v,p)
v = 127467388                   tB = 37944166
```

2.4. Pasirašyti t_B su Šnoro parašu $\sigma_B = (r_B, s_B)$ pasirinkus atsitiktinį skaičių $z \leftarrow randi Z_{p-1}$), kad $1 < z < p-1$:

2.4.1. Apskaičiuoti pirmą parašo komponentę: $r_B = g^z \bmod p$:

```
>> z=int64(randi(p-1))          >> rB=mod_exp(g,z,p)
z = 35145884                    rB = 142067410
>> 1 < z & z < p-1
ans = 1
```

2.4.2. Sujungti (angl. concat) t_B ir r_B, apskaičiuojant santrauką h_B=H(t_B||r_B) ir apskaičiuoti antrąją parašo komponentę $s_B = (z + yh) \bmod (p - 1)$:

```
>> hB = hd28(concat(tB, rB))    >> sB=mod((z+y*hB),p-1)
hB = 44653728                   sB = 102969376
```

2.4.3. Parašas h_B santraukai yra $\sigma_B = (r_B, s_B)$
Pasirašyti(y, h_B) = $\sigma_B = (r_B, s_B)$ = (142067410, 102969376).

2.5. Bankas siunčia **Aldonai** savo viešą raktą b, sesijos viešą parametrą t_B ir šiam parametrui suformuotą parašą σ_B.

Aldonos pinigų pervedimo operacijos pranešimo parengimą ir perdavimą į **Banką** vaizduojanti schema pateikiama 3 pav.

a, t_A, r_A, s_A

b, t_B, r_B, s_B

3.1 Apskaičiuoti h' ir patikrinti:

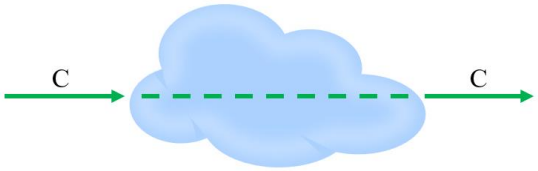
$$h' = H(t_B \| r_B),$$

$$g^{s_B} \bmod p = r_B\, b^{h'} \bmod p.$$

V1 = V2

3.2 Apskaičiuoti bendrą slaptą simetrinį raktą k:

$$k = t_B{}^u \bmod p$$



3.3 Transformuoti k į šešioliktainę formą:

k_h = hexadecimal(k);

3.4 Pasirinkti pranešimą m (*pvz.*, m = '1020') ir užšifruoti:

C = AES128(m, k_h, 1, 'e').

3 pav. Aldona siunčia pranešimą į Banką

Aldonos veiksmai siunčiant pinigų pervedimo operacijos pranešimą į **Banką**:

3.1. Apskaičiuoti santrauką h' = H(t_B || r_B), **V1** = $g^{sB} \bmod p$, **V2** = $r_B\, b^{h'} \bmod p$ ir patikrinti ar **V1 = V2**:

V1

>> V1=mod_exp(g,sB,p)

V1 = 99127062

V2

>> htrB=hd28(concat(tB,rB))

htrB = 44653728

>> b_htrB=mod_exp(b,htrB,p)

b_htrB = 1368610

>> V2=mod(rB*b_htrB,p)

V2 = 99127062

>> V1==V2

ans = 1 ← jeigu **1** parašas tikras

3.2. Apskaičiuoti bendrą slaptą simetrinį raktą $k = t_B{}^u \bmod p$:

>> k=mod_exp(tB,u,p)

k = 137071686

3.3. Transformuoti k į šešioliktainę formą: k_h = hex(k):

>> kh=dec2hex(k, 32)

kh = 00000000000000000000000082B8C46

3.4. Pasirinkti pranešimą m (*pvz.*, m = '1020') ir užšifruoti:

>> NR=1

NR = 1

>> fun='e'

fun = e

>> m='1020'

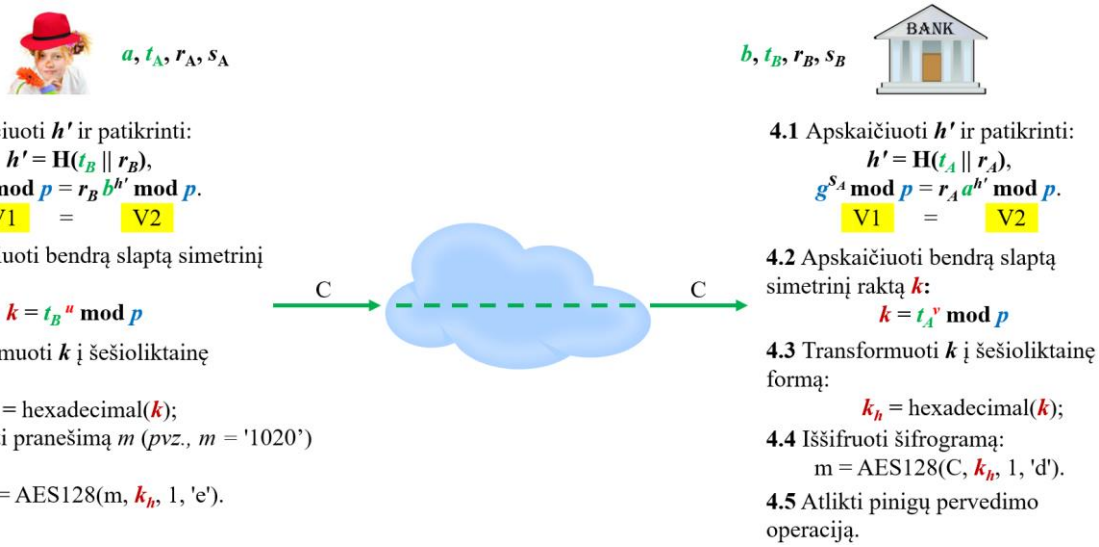
m = 1020

>> c = AES128(m, kh, NR, fun)

c = 930739089307f45393cb3953ea2cb515

3.5. Aldona siunčia **Bankui** šifrogramą **c**.

Banko pranešimo priėmimą vaizduojanti schema pateikiama 4 pav.



4 pav. Bankas peržiūri Aldonos atsiųstą pranešimą

Banko veiksmai į **Aldonos** pranešimą:

4.1. Apskaičiuoti santrauką $h' = H(t_A \| r_A)$, $V1 = g^{sA} \bmod p$, $V2 = r_A\, a^{h'} \bmod p$ ir patikrinti ar **V1 = V2**:

V1

>> V1=mod_exp(g,sA,p)

V1 = 110503357

V2

>> htrA=hd28(concat(tA,rA))

htrA = 96291913

>> a_htrA=mod_exp(a,htrA,p)

a_htrA = 51206859

>> V2=mod(rA*a_htrA,p)

V2 = 110503357

>> V1==V2

ans = 1 ← jeigu **1** parašas tikras

4.2. Apskaičiuoti bendrą slaptą simetrinį raktą $k = t_A^v \bmod p$:

>> k=mod_exp(tA,v,p)

k = 137071686

4.3. Transformuoti k į šešioliktainę formą: k_h = hex(k):

>> kh=dec2hex(k, 32)

kh = 00000000000000000000000082B8C46

4.4. Iššifruoti šifrogramą **c**:

>> NR=1

NR = 1

>> fun='d'

fun = d

>> ms = AES128(c, kh, NR, fun)

ms = 1020

4.10. Bankas atlieka pinigų ***ms*** pervedimo operaciją.

Užduotys Minimaliam-HTTTPS.

Užduotims naudojami viešieji parametrai p=int64(144668519); g=7.

1. Turėdami **Aldonos** privatų raktą x, atsitiktinius skaičius u ir i, nustatykite, kuriam viešam sesijos parametrui t_A ir šiam parametrui suformuoto parašo $\mathbf{б_A} = (r_A, s_A)$ apskaičiavimui buvo panaudotos šios reikšmės:
   1. x=int64(123650908), u=int64(80848762), i=int64(130251774);
   2. x=int64(57298159), u=int64(95805496), i=int64(20908711);
   3. x=int64(115409770), u=int64(54139791), i=int64(47507127);
   4. x=int64(130694878), u=int64(45252398), i=int64(18701868).

   Vieši sesijos parametrai t_A ir jiems suformuoti parašai $\mathbf{б_A} = (r_A, s_A)$:

   1. t_A = 107126995, r_A = 95681387, s_A = 85220607;
   2. t_A = 37453652, r_A = 35107067, s_A = 66096703;
   3. t_A = 65652735, r_A = 26475295, s_A = 66096703;
   4. t_A = 107126995, r_A = 95681387, s_A = 124870334;
   5. t_A = 37453652, r_A = 35107067, s_A = 11477016;
   6. t_A = 143285385, r_A = 126735887, s_A = 85220607;
   7. t_A = 65652735, r_A = 26475295, s_A = 11477016;
   8. t_A = 143285385, r_A = 126735887, s_A = 124870334.

2. Turėdami **Aldonos** atsitiktinį skaičių u ir pervedamą pinigų sumą m, **Banko** viešą raktą b, viešą sesijos parametrą t_B, parašą $\mathbf{б_B} = (r_B, s_B)$, nustatykite, ar **Banko** parašas viešam sesijos parametrui galioja ir kuri **Aldonos** šifrograma **c** suformuota pinigų sumai m, naudojantis šiomis pateiktomis reikšmėmis:
   1. b=int64(33290112), u=int64(13818701), t_B=int64(22393260), r_B=int64(133342595), s_B=int64(48486702), m="1935";
   2. b=int64(124518796), u=int64(63490768), t_B=int64(132211809), r_B=int64(62174403), s_B=int64(101331856), m="15";
   3. b=int64(89264145), u=int64(10002329), t_B=int64(144413333), r_B=int64(125706375), s_B=int64(26860092), m="625";
   4. b=int64(143855886), u=int64(55412227), t_B=int64(65755586), r_B=int64(35350173), s_B=int64(13118777), m="58".

   Parašų galiojimas ir šifrogramos **c**:

   1. **Parašas galioja,** **c** = "cb31833acb311f1fcbfb831fd586cbbf ";
   2. **Parašas negalioja,** **c** = "6db3a4376db3c7a66d6da4a6cc48c497";
   3. **Parašas galioja,** **c** = "9219290b92195391924b2991630e94cd";
   4. **Parašas galioja,** **c** = "464e9fac464ee31f46099f1f58e8654a".

3. Turėdami **Banko** atsitiktinį skaičių v, **Aldonos** viešą raktą a, viešą sesijos parametrą t_A, parašą $\sigma_A = (r_A, s_A)$, šifrogramą **c** nustatykite, ar **Aldonos** parašas viešam sesijos parametrui galioja ir kuri **Aldonos** pervedama pinigų sumą m atitinka šifrogramą **c**, naudojantis šiomis pateiktomis reikšmėmis:

1. v=int64(96251085), a=int64(2673728), t_A=int64(92674994), r_A=int64(135957744), s_A=int64(97243985), **c**="563b4fb5563b840c566c4f0c5c13114e";
2. v=int64(33121508), a=int64(106628859), t_A=int64(12260360), r_A=int64(12260360), s_A=int64(77101193), **c**="2cc192062cc100182c35921837778833";
3. v=int64(39771512), a=int64(55854671), t_A=int64(82187889), r_A=int64(105460384), s_A=int64(133523331), **c**="ae0e7960ae0e8d91aea479915f29cad2";
4. v=int64(58307924), a=int64(92612014), t_A=int64(95778795), r_A=int64(55750710), s_A=int64(127980008), **c**="3c51d1143c51f4183c6cd11827cc7026".

Parašų galiojimas ir pinigų sumos m:

1. **Parašas galioja,** m = 318;
2. **Parašas negalioja,** m = 98;
3. **Parašas galioja,** m = 1567;
4. **Parašas negalioja,** m = 18.

Minimalus-Https, Imimsociety svetainės užduotis

Imimsociety Mini-HTTPS uždavinio 3 dalyje (žr. 5 pav.) turite įvesti į Octave mentoriaus atsiųstą viešą sesijos parametrą t_B (kaip t_B kintamojo vertę naudoti K_B vertę).

3. Mentor sends you (t_B=32768, K_B =136143786, R =105173192, S =2353771). Verify Mentor's signature $\sigma_M = (R, S)$ on t_B. If signatur is valid then taking S compute verification parameter $V_1 = g^S \ mod \ p$. Compute common symmetric secret key k and transform k to the hexadecimal form k_h of 32 digits length as it is required for AES128 function. Create the string of message variable m ='MMDD' consisting of the month and day of your birth. Encrypt message m using 1 round of AES128 cipher with key k_h by computing ciphertext >> C=AES128(m,Kh,1,'e'). Attention! Encryption using 1 round is extremely insecure and is used there to speed up the computations and to make sure of its insecurity. Insecurity is seen by comparing plaintext and ciphertext messages in hexadecimal format. They have non-excrypted digits.C should be entered within ″.
Send $[V_1, C]$ to the Mentor for decryption.

125256920, "1f9df0bd1f9db4141fd8f014bd8efc36"

5 pav. Imimsociety Mini-HTTPS uždavinio 3 dalis