Proxy Signature

```
>> Soft='Software created by Alice company Alice and Co'
Soft = Software created by ALice company Alice and Co
```

```
>> p
p = int64(268435019)
>> g = 2;
>> x=int64(randi(p-1))
x = 132355164
>> a=mod_exp(g,x,p)
a = 133074594
```

```
>> t=int64(randi(p-1))
t = 122446162
>> b=mod_exp(g,t,p)
b = 84002361
>> tb=mod(t*b,p-1)
tb = 116557960
>> y=mod(x+tb,p-1)
y = 248913124

>> y= mod(x+t*b,p-1)
```

```
>> g_y=mod_exp(g,y,p)
g_y = 127311302
>> b_b=mod_exp(b,b,p)
b_b = 42714980
>> ab_b=mod(a*b_b,p)
ab_b = 127311302
```

```
>> h=hd28('Soft') % Error
h = 252284385
>> h=hd28(Soft)  % Good
h = 141643577
>> ksy=int64(randi(p-1))
ksy = 248849331
>> r=mod_exp(g,ksy,p)
r = 95766604
>> yh=mod(y*h,p-1)
yh = 58196966
>> s=mod(ksy+yh,p-1)
s = 38611279

>> s=mod(ksy+y*h,p-1)
```

User's verificatio of Bob's signature on the Soft

```
>> g_s=mod_exp(g,s,p)
g_s = 179785879

>> b_b=mod_exp(b,b,p)
b_b = 42714980

>> ab_b=mod(a*b_b,p)
ab_b = 127311302

>> ab_b_h=mod_exp(ab_b,h,p)
ab_b_h = 1353047
>> rab_b_h=mod(r*ab_b_h,p)
rab_b_h = 179785879
```

Subliminal Channel - Steganography using ElGamal Signature

```
>>  p =   int64(268435019)
p = 268435019
>> g=2;
>> z=int64(randi(p-1))
z = 7688965
>> c=mod_exp(g,z,p)
c = 217126641
```

```
>> M='Bob I wait you'
M = Bob I wait you
>> h=hd28(M)
h = 240745341
```

```
>> s=int64(126312337)
s = 126312337
>> gcd(s,p-1)
ans = 1
```

```
>> s_m1=int64(212931827)
s_m1 = 212931827
>> mod(s*s_m1,p-1)
ans = 1
```

Subliminal Channel - Steganography Using Schnorr Signature

M - masking message to be signed.

gcd(**z**, **p**-1) = 1, then **z** $^{-1}$ mod(**p**-1) exists.

Using Schnorr Signature

M - masking message to be signed.
m' = k - secret message to be sent.

$r = g^k \bmod p$
$h = H(M \| r)$
$s = k + z * h \bmod (p-1)$

$M, \sigma = (r, s) \rightarrow$ B: $h = H(M \| r)$

$V1 = g^s \bmod p$
$V2 = r * c^h \bmod p$
$V1 \stackrel{?}{=} V2$

Corrected

$k = (s - z * h) \bmod (p-1) = m'$

gcd(**z**, **p**-1) = 1, then **z** $^{-1}$ mod(**p**-1) exists.

>> p = int64(268435019)
p = 268435019
>> g=2;
>> z=int64(randi(p-1))
z = 7688965
>> c=mod_exp(g,z,p)
c = 217126641

>> M='Bob I wait you impatiently'
M = Bob I wait you
>> h=hd28(M)
h = 240745341

>> k=int64(randi(p-1))
k = 21398217
>> r=mod_exp(g,k,p)
r = 92029919
>> s=mod(k+z*h,p-1)
s = 34241676

>> V1=mod_exp(g,s,p)
V1 = 63032245
>> c_h=mod_exp(c,h,p)
c_h = 62405921
>> V2=mod(r*c_h,p)
V2 = 63032245

>> kk=mod(s-z*h,p-1)
kk = **21398217**

Bit Commitment using RSA (partially)

Key Generation

Public Parameter $PP = (p)$ p - may be strong prime

A: $K_A = (e_A, d_A)$
$e_A \cdot d_A = 1 \bmod (p-1)$

B: $K_B = (e_B, d_B)$ % private keys
$e_B \cdot d_B = 1 \bmod (p-1)$

$e_A \leftarrow randi : gcd(e_A, p-1) = 1 \Rightarrow \exists! \; d_A = e_A^{-1} \bmod (p-1)$

$d_A = e_A^{-1} \bmod p$

$d_B = e_B^{-1} \bmod p$

>> p
p = 268435019
>> eA=int64(randi(p-1))
eA = 113108923
>> gcd(eA,p-1)
ans = 1

>> eB=int64(randi(p-1))
eB = 175845008
>> gcd(eB,p-1)
ans = 2
>> eB=int64(randi(p-1))
eB = 213299919

eA = 113108923
>> gcd(eA,p-1)
ans = 1
>> dA=mulinv(eA,p-1)
dA = 114348687
>> mod(eA*dA,p-1)
ans = 1

ans = 2
>> eB=int64(randi(p-1))
eB = 213299919
>> gcd(eB,p-1)
ans = 1
>> dB=mulinv(eB,p-1)
dB = 182812595
>> mod(eB*dB,p-1)
ans = 1

Till this place