1. Assume that in the definition of ELGamal signatures we do not hash the message to be signed and therefore the equation defining s component of a signature on M is as follows:

$$s := (M - x \cdot r)/k \bmod (p-1)$$

(x is the private key). Then select any (u, v) such that $GCD(v, p-1) = 1$. Take $r = g^u X^v \bmod p$ and $s = -r/v \bmod (p-1)$.

Show that (r, s) is a valid signature for $s \cdot u \bmod (p-1)$. $= M$

Weryfikacja podpisu: 1° Sprawdź, czy $0<r<p$ i $0<s<p-1$

2° Sprawdź, czy $g^M \equiv X^r r^s \pmod p$

1° $0<r<p$, bo $r = g^u X^v \bmod p$. (g jest generatorem, więc nie jest elementem zerowym)
$0<s<p-1$, bo $s = -r/v \bmod (p-1)$ i $r \neq 0$.

2° $L = g^M = g^{s \cdot u}$

$P = X^r r^s = X^r (g^u X^v)^s = X^r \cdot g^{u \cdot s} \cdot (X^v)^{-r/v} = g^{s \cdot u} \cdot X^r \cdot X^{-r} = g^{s \cdot u}$ □

2. The verification procedure for ElGamal involves checking that $r \in [1, p-1]$. Show that this is necessary.

Hint: take any valid signature (r, s), say for a message m. Take any message m' and $u = \frac{\text{Hash}(m')}{\text{Hash}(m)} \bmod (p-1)$. Take $s' := s \cdot u \bmod (p-1)$ and r' such that $r' = r \cdot u \bmod (p-1)$ and $r' = r \bmod p$ (use Chinese Remainder Theorem to calculate r'). Consider (r', s').

Why checking the range of r helps? Should we do such range checking for DSA and Schnorr?

Weryfikacja podpisu: $g^{H(m)} \equiv X^r r^s \pmod p$

$L = g^{H(m')}$

$P = X^{r'} r'^{s'} \overset{1^\circ}{=} X^{r'} r^{s'} \overset{2^\circ}{=} X^{r \cdot u} \cdot r^{s \cdot u} = (X^r \cdot r^s)^u = (X^r \cdot r^s)^{\frac{H(m')}{H(m)}} = (g^{H(m)})^{\frac{H(m')}{H(m)}} = g^{H(m)} = L$

1° podstawienie $r' \equiv r \bmod p$; r' użyte jako element grupy $\mathbb{Z}_p^*$

2° podstawienie $r' \equiv r \cdot u \bmod (p-1)$; r' użyte jako wykładnik w $\mathbb{Z}_p^*$

Range check: $0 < r < p$: dla $\begin{cases} r' \equiv r \bmod p \\ r' \equiv r \cdot u \bmod (p-1) \end{cases}$ w takim zakresie mamy tylko jedno

rozwiązanie $\begin{cases} r' = r \\ u = 1 \end{cases}$, z czego wynika, że $H(m) \equiv H(m') \bmod (p-1)$.

W schemacie Schnorra nie trzeba robić takiego sprawdzenia, bo nie ma zmiennych, które są jednocześnie używane jako element grupy w podstawie i element grupy w wykładniku.

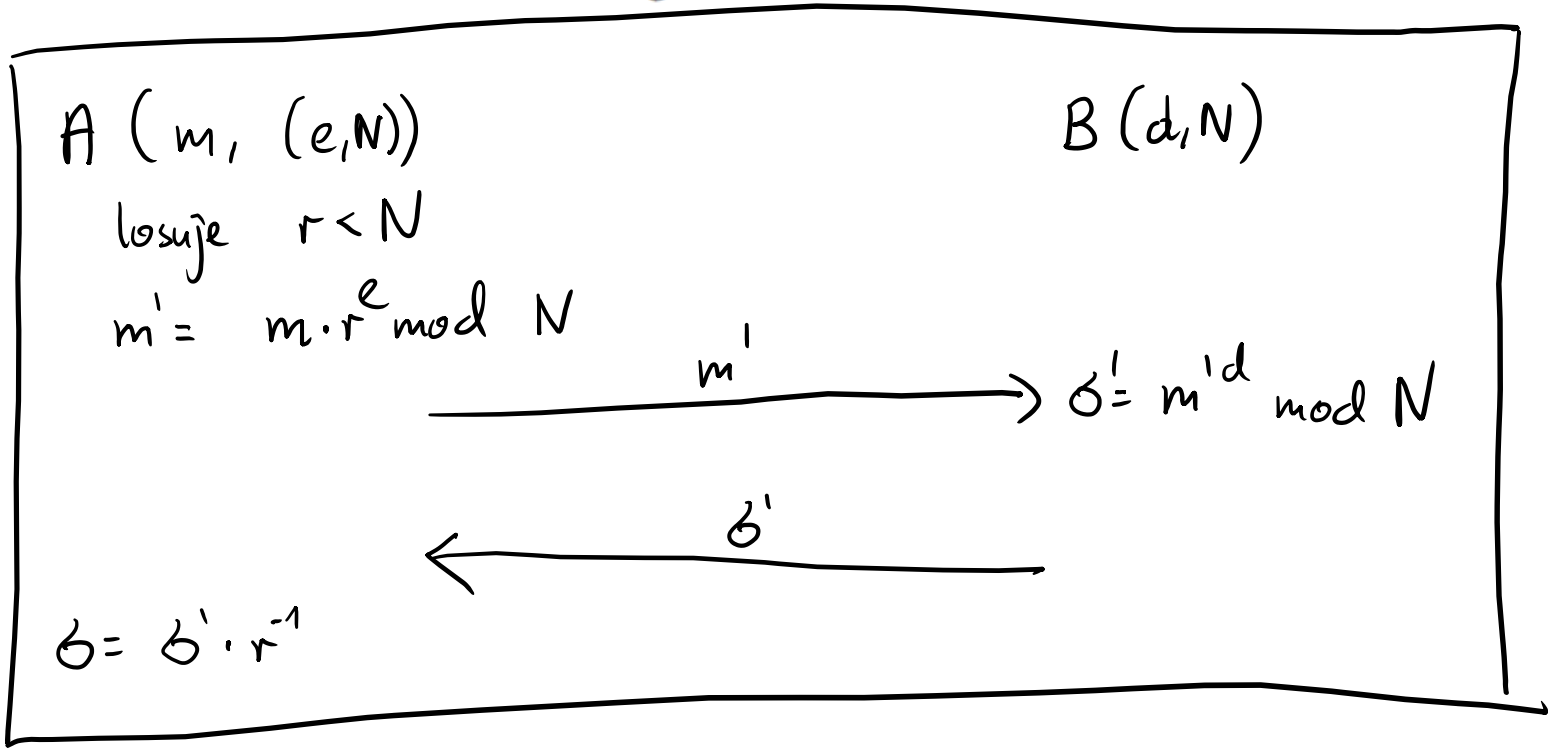
O ile sam algorytm DSA wymaga sprawdzania, to

1) dla r nie ma to znaczenia, r musi być równy $v=((\dots) \bmod p) \bmod q$, więc r i tak musi być mniejsze od q (natomiast trzeba sprawdzić, czy $r \neq 0$).
2) dla s jest podobnie jak w schemacie Schnorra, przy czym dla samego algorytmu sprawdzenia s musi być odwracalny, więc $s \neq 0 \cdot \bmod q$.

3. Sometimes we need a blind signature: Alice presents a sealed envelope to Bob. Inside the envelope there is a carbon paper, so when Bob signs the envelope the signature is copied to the document inside the envelope. The signature is called blind, as Bob does not know what he is signing.

   Blind signatures can be used for instance in voting: the voter comes with a ballot in the the envelope, election official signs blindly, the voter opens the envelope and puts the signed ballot to the ballot box.

   Your task: convert RSA to a blind signature scheme.



$sk = (d, N)$
$pk = (e, N)$

(zakładamy, że m jest już zakodowaną wiadomością, np. zobacz PKCS#1, sekcja "Encoding methods for signatures with appendix")

Weryfikacja podpisu σ dla m przechodzi, bo $\sigma = \sigma' \cdot r^{-1} = m'^d r^{-1} \bmod N = (m \cdot r^e \bmod N)^d r^{-1} \bmod N =$
$= m^d \cdot r^{ed} \cdot r^{-1} \bmod N = m^d \bmod N$. ($ed = 1$)

4. Generating the private key for creating signatures is a critical issue. If you get the private key x from your service provider, then the provider may retain the key and later forge your signatures. On the other hand, if you create your private key yourself, then you depend on random numbers generated by your computer. Both options are risky.

   The solution is that you get a "prekey" x_0 from your provider and choose your share x_1. Finally, your private key is $x_0 + x_1 \bmod q$.

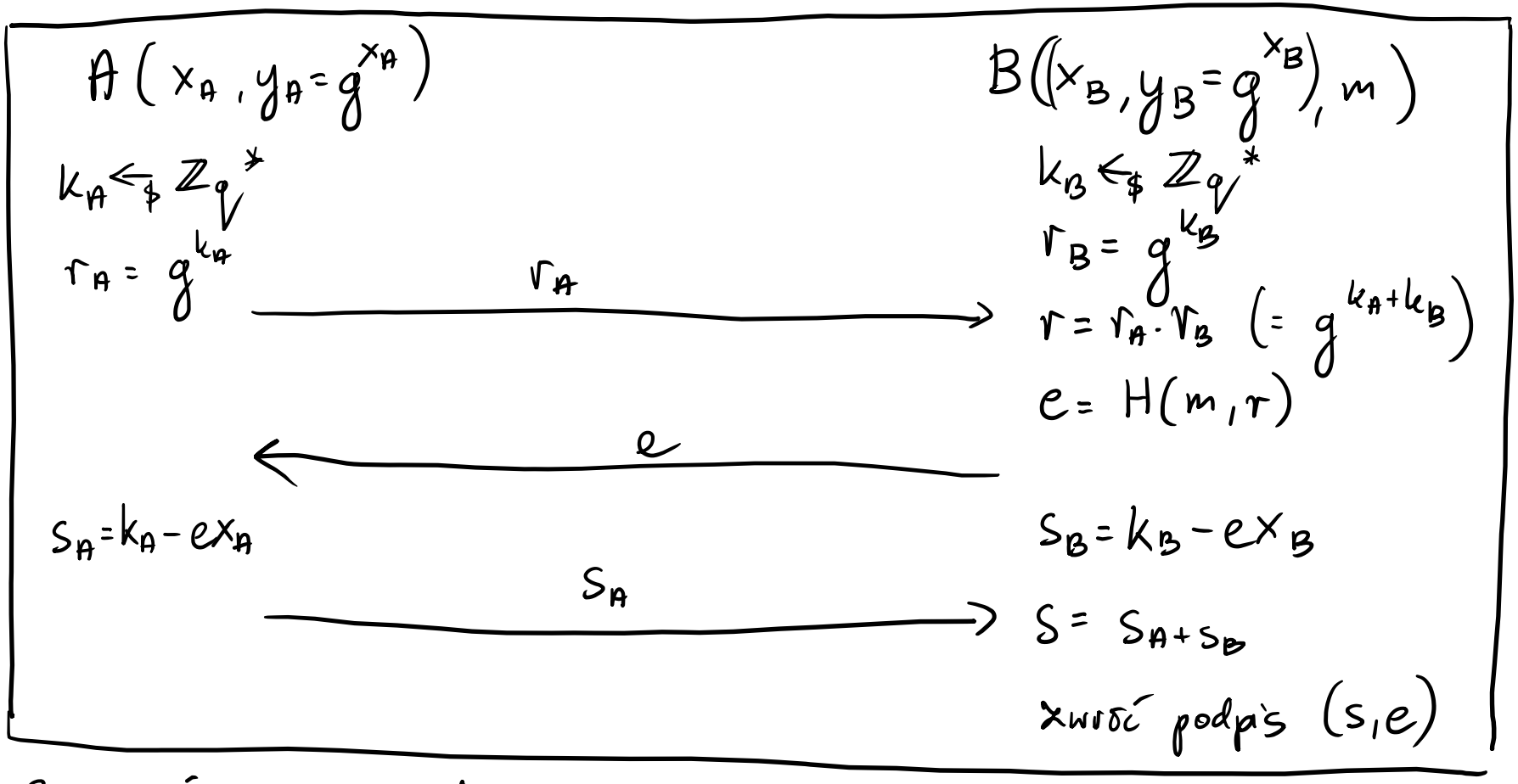
   Formulate all details of a protocol based on this idea (all messages exchanged and calculations on both sides). Make sure that neither the provider nor your computer will be able to determine the final result of his choice. Make sure that the provider does not learn the final private key.

TBD

5. Why it is impossible to apply a similar idea for RSA?

TBD

6. Since a secret key may be leaked from the signing device let us split the private key so that there are two shares: one on device A, and one on device B. In order to create a signature, devices A and B should cooperate. How to do it for the Schnorr signatures?



Cel: złożyć podpis na m, który zweryfikuje się kluczem $y = g^{x_A + x_B}$, Podpis musi być zrobiony tak, żeby w przyszłości ani A, ani B nie mogło samodzielnie tworzyć podpisów dla klucza y.

Bezpieczeństwo sprowadza się do schematu identyfikacji Schnorra (A identyfikuje się względem B).

$\text{Verify}_y((s,e), m)$:

$e \stackrel{?}{=} H(m, g^s \cdot y^e)$ // $g^s \cdot y^e = g^{s_A + s_B} \cdot g^{x \cdot e} = g^{k_A - e x_A + k_B - e x_B} \cdot g^{(k_A + k_B) e} = g^{k_A + k_B} = r$