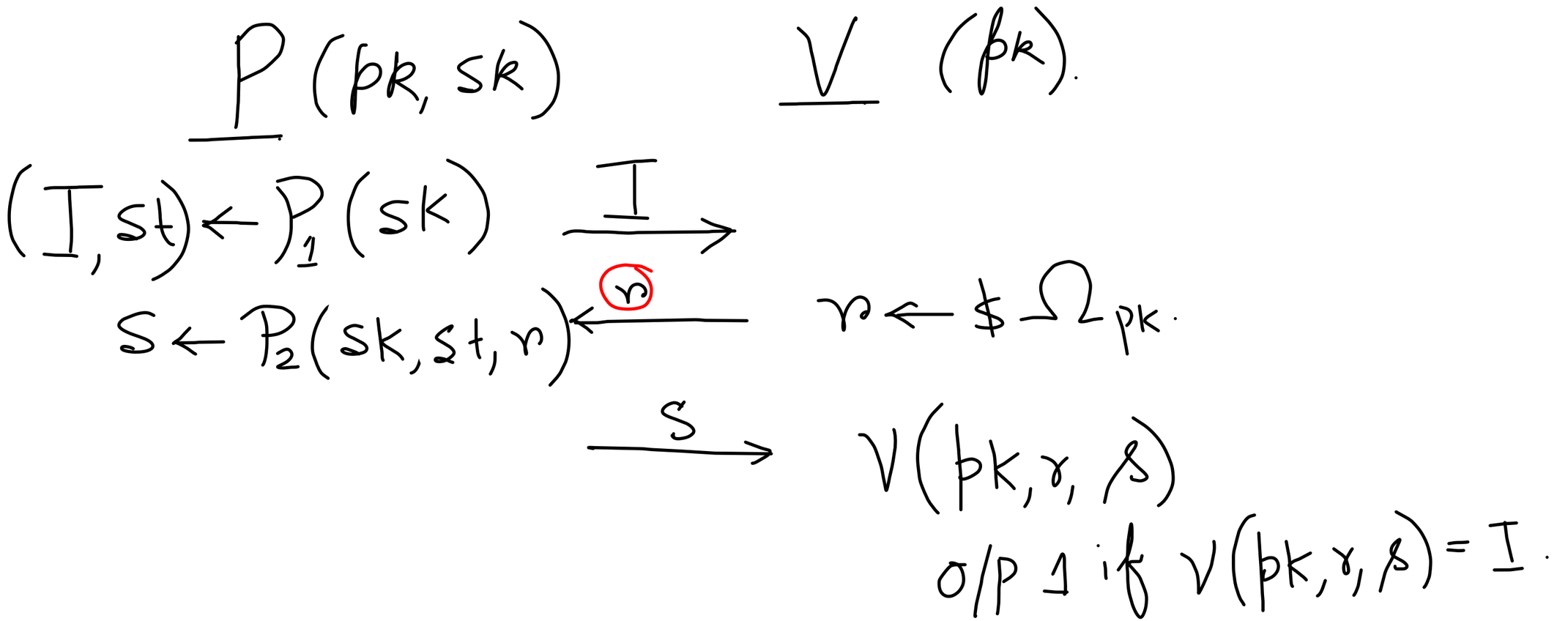
Identification Scheme.



Fiat-Shamir Transformation.

Signature algorithm:

1. KeyGen: on i/p 1^n, it outputs (pk, sk) and H is a hash fn which is implicitly defined.

2. Sign: on message m, signer will compute $(I, st) \leftarrow P_1(sk)$. and then it computes $r \leftarrow H(I, m)$. and then it computes $s \leftarrow P_2(sk, st, r)$ and outputs (r, s)

3. Verification on i/p $(pk, m, (r, s))$, it outputs 1 if and only if

(i) $I \leftarrow V(pK, r, s)$ and $H(I, m) \stackrel{?}{=} r$.

V → part of identification scheme

Correctness: $\forall (pk, sk)$ $\forall$ message m

$$Verify(pk, m, Sign(sk, m)) = 1.$$

Schnorr Signature scheme.

KeyGen (1^n) = pk, sk

$pk = G, q, g, g^x = y$ $\quad x \xleftarrow{\$} Z_q$

$sk = x$

$I = g^k$

Sign(sk, m) $\quad P_1(x) \rightarrow (g^k, k)$ $\quad k \xleftarrow{\$} Z_q$

$H(g^k, m) \longrightarrow r$ $\quad s \leftarrow rx + k \bmod q$

(r, s) $\quad V(pk, (r, s), m)$ -

Schnorr Signature scheme.

KeyGen (1^n) = pk, sk

$pk = G, q, g, g^x = y$ $\quad x \xleftarrow{\$} Z_q$

$sk = x$

$I = g^k$

$P_1(x) \rightarrow (g^k, k)$ $\quad k \xleftarrow{\$} Z_q$

Sign(sk, m)

$s \leftarrow rx + k \bmod q$

$H(g^k, m) \longrightarrow r$

$V(r \quad (r, s)$ Verify$(pk, (r, s), m)$ -

$$V(r, s, pk) \rightarrow I = g^s y^{-r}$$

$$H(g^k, m) \stackrel{?}{=} r$$

Thm: If the identification scheme is secure and H is modelled as random oracle, then the signature scheme obtained by Fiat-Shamir transformation is unforgeable.

Let A be the forging algorithm that breaks the signature scheme $\bar{\Pi}$.

We construct an adversary A' that breaks the identification scheme $\bar{\Pi}'$.

Let $q(n)$ be the upper bound on the number of random oracle queries.

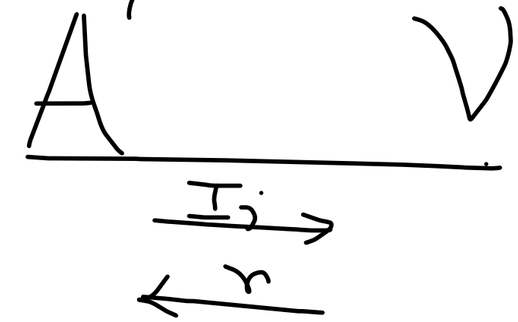
Assumptions:

(i) all the random oracle queries has to be distinct

(ii) If adversary obtains a signature (r, s) on message query m, with $V(pk, r, s) = 1$, then the adversary will not make any random oracle query with i/p $(1, m)$

(iii) If adversary outputs a forgery $(m^*, (r^*, s^*))$ with $V(pk, r^*, s^*) = 1$, then adversary must have a prior query to the random oracle with i/p $(1, m^*)$

Random oracle query.



$A \rightarrow H(I_\alpha, m_\alpha)$, $\alpha = 1(1)q$.

if $\alpha = j$, A' sends I_j to its verifier and it receives r. A' sends that r to A and set $H(I_j, m_j)$ to r.

if $\alpha \neq j$, then A' randomly samples r and send it to A, and set $H(I_\alpha, m_\alpha)$ to r.

Signing oracle query.

A makes query of the form m.

A' will query to $Trans_{SK}$ oracle and it receives (I, r, s) A' sends (r, s) to A.

Forging: A outputs $(m^*, (r^*, s^*))$ compute $I^* \leftarrow V(pk, r^*, s^*)$

Check whether $(I_j, m_j) = (I^*, m^*)$ If it is output s^*.

Signing oracle query.

A makes query of the form m. $H(I, m) = r'$

A' will query to $Trans_{SK}$ oracle and it receives (I, r, s) A' sends (r, s) to A.

Forging: A outputs $(m^*, (r^*, s^*))$ compute $I^* \leftarrow v(pk, r^*, s^*)$

Check whether $(I_j, m_j) = (I^*, m^*)$ If it is output s^*.

Reduction Game

1. choose $j \leftarrow \$ \{1, \dots, q\}$.