```
HI1LEMFOperations DEFINITIONS AUTOMATIC TAGS ::=
BEGIN
IMPORTS
  LawfulInterceptionIdentifier,
  TimeStamp
FROM
  UnsignedRequestDetail;

HI1LEMFPDU ::= SEQUENCE
{
  version [0] Version,
  content [1] Content,
  operator [2] UTF8String OPTIONAL,
  ...
}

Version ::= ENUMERATED
{
  version1 (1),
  ...
}

Content ::= CHOICE
{
  request [1] Request,
  respond [2] Respond,
  ...
}

SignedRequest ::= SEQUENCE
{
  version [1] SignedRequestVersion,
  signStandard [2] OBJECT IDENTIFIER,
  -- CryptographicMessageSyntax2004 { iso(1) member-body(2) us(840) rsadsi(113549)
pkcs(1) pkcs-9(9) smime(16) modules(0) cms-2004(24)
  cmsDERSignedRequest     [3] OCTET STRING,
  -- cmsDERSignedRequest [3] ANY DEFINED BY signStandard
  ...
}
```

```
Request ::= CHOICE
{
  simpleRequest [1] SimpleRequest,
  signedRequest [2] SignedRequest,
  ...
}

SimpleRequest ::= CHOICE
{
  helloRequest [1] HelloRequest,
  listRequest [2] ListRequest,
  rtRequest [3] RTRequest, -- Żądanie włączenia lub  wyłączenia trybu online
(dotyczą tylko obserwacji rozpoczętych) dla obserwacji aktywnych
  ...
}
RTRequest ::= SEQUENCE
{
  liid [1] LawfulInterceptionIdentifier,
  action [2] ENUMERATED
  {
    start(0), -- Włączenie odsłuchu online
    stop(1) -- Wyłączenie odsłuchu online
  },
}

SignedRequestVersion ::= ENUMERATED
{
  v1 (0),
  ...
}

HelloRequest ::= SEQUENCE
{
  message [1] UTF8String,
  ...
}

Respond ::= CHOICE
{
  generalRespond [1] GeneralRespond,
  listRespond [2] ListRespond,
  ...
}
```

```
GeneralRespond ::= SEQUENCE
{
  result [1] Result,
  message [2] UTF8String OPTIONAL,
  -- return Hello request message
  ...
}

Result ::= ENUMERATED
{
  ok (1),
  missing-parameter (2),
  unknown-parameter (3),
  unknown-parameter-value (4),
  incorrect-BER (5),
  badSignature (6),
  certificateExpired (7),
  unknownError (10),
  unsupportedService (11),
  ...
}

CheckRespond ::= SEQUENCE
{
  liid [1] LawfulInterceptionIdentifier,
  checkStatus [2] CheckStatus,
  message [3] UTF8String OPTIONAL,
  ...
}

CheckStatus ::= ENUMERATED
{
  notFound (0),
  waiting (1),  -- założone przez lemf, nie ma w cn (czeka na zatwierdzenie lub )
  cnActivated (2), -- jest w cn
  unknown (3),
  deActivated (4), -- po deaktywowaniu w cn
  ...
}

ListRequest ::= SEQUENCE
{
```

```
  type [1] ListType,
  liid [2] LawfulInterceptionIdentifier OPTIONAL,
  ...
}

ListRespond ::= SET OF CheckRespond

ListType ::= ENUMERATED
{
  all (1),
  specific (2),
  ...
}

END
```