The new Internet Protocol security IPsec testing with TTCN-3

Ariel Sabiguero [1,2] María Eugenia Corti[1] César Viho[2]

[1]Instituto de Computación, Facultad de Ingeniería, Universidad de la República
J. Herrera y Reissig 565, Montevideo, Uruguay
{asabigue,mcorti}@fing.edu.uy

[2]IRISA / Dionysos
Campus Universitaire de Beaulieu
35042 Rennes CEDEX, France
{asabigue,viho}@irisa.fr

30/05/2007

Suite of security protocols

	Authentication Header (AH)	Encapsulating Security Payload (ESP)
Connectionless Integrity	√	√

Suite of security protocols

	Authentication Header (AH)	Encapsulating Security Payload (ESP)
Connectionless Integrity	√	√
Data Origin Authentication	√	√

Suite of security protocols

	Authentication Header (AH)	Encapsulating Security Payload (ESP)
Connectionless Integrity	√	√
Data Origin Authentication	√	√
Access Control	√	√

Suite of security protocols

	Authentication Header (AH)	Encapsulating Security Payload (ESP)
Connectionless Integrity	✓	✓
Data Origin Authentication	✓	✓
Access Control	✓	✓
Confidentiality	✗	✓

Set of cryptographic algorithms

Encryption algorithm

- 3DES-CBC

Authentication algorithm

Set of cryptographic algorithms

Encryption algorithm

- 3DES-CBC
- NULL

Authentication algorithm

Set of cryptographic algorithms

Encryption algorithm

- 3DES-CBC
- NULL
- AES-CBC

Authentication algorithm

Set of cryptographic algorithms

Encryption algorithm

- 3DES-CBC
- NULL
- AES-CBC
- AES-CTR

Authentication algorithm

Set of cryptographic algorithms

Encryption algorithm

- 3DES-CBC
- NULL
- AES-CBC
- AES-CTR

Authentication algorithm

- HMAC-SHA1-96

Set of cryptographic algorithms

Encryption algorithm

- 3DES-CBC
- NULL
- AES-CBC
- AES-CTR

Authentication algorithm

- HMAC-SHA1-96
- NULL

Set of cryptographic algorithms

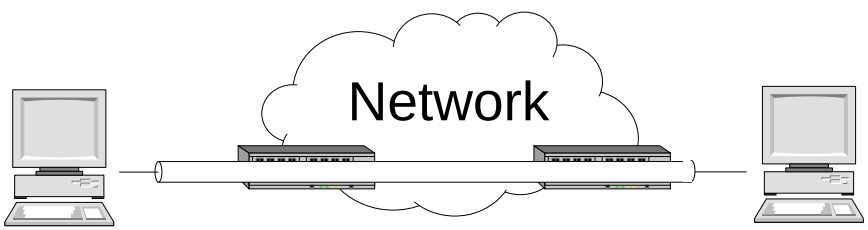
Encryption algorithm

- 3DES-CBC
- NULL
- AES-CBC
- AES-CTR

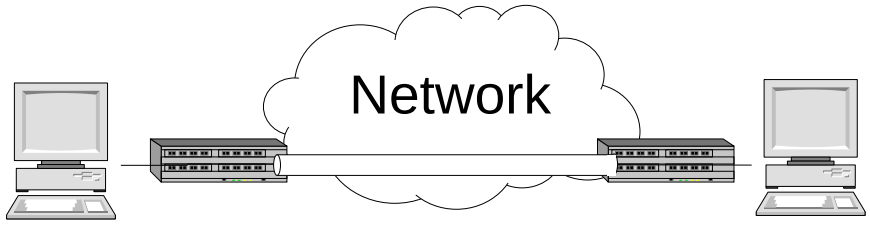
Authentication algorithm

- HMAC-SHA1-96
- NULL
- AES-XCBX-MAC-96

IPsec modes



Transport mode



Tunnel mode

SPD and SA

Security Policy Database

- control IPsec traffic

SPD and SA

Security Policy Database

- control IPsec traffic
- consulted for incoming and outgoing traffic

SPD and SA

Security Policy Database

- control IPsec traffic
- consulted for incoming and outgoing traffic

Security Association

SPD and SA

Security Policy Database

- control IPsec traffic
- consulted for incoming and outgoing traffic

Security Association

- simplex "connection" that affords security services to the traffic carried by it.

SPD and SA

Security Policy Database

- control IPsec traffic
- consulted for incoming and outgoing traffic

Security Association

- simplex "connection" that affords security services to the traffic carried by it.
- each SA an entry in the SA Database (SAD)

SPD and SA

Security Policy Database

- control IPsec traffic
- consulted for incoming and outgoing traffic

Security Association

- simplex "connection" that affords security services to the traffic carried by it.
- each SA an entry in the SA Database (SAD)
- one SA for each traffic direction

v6RL test suite coverage

- Tunnel and Transport mode

v6RL test suite coverage

- Tunnel and Transport mode
- A combination of authentication and encryption algorithms

v6RL test suite coverage

- Tunnel and Transport mode
- A combination of authentication and encryption algorithms
- Only ESP

v6RL test suite coverage

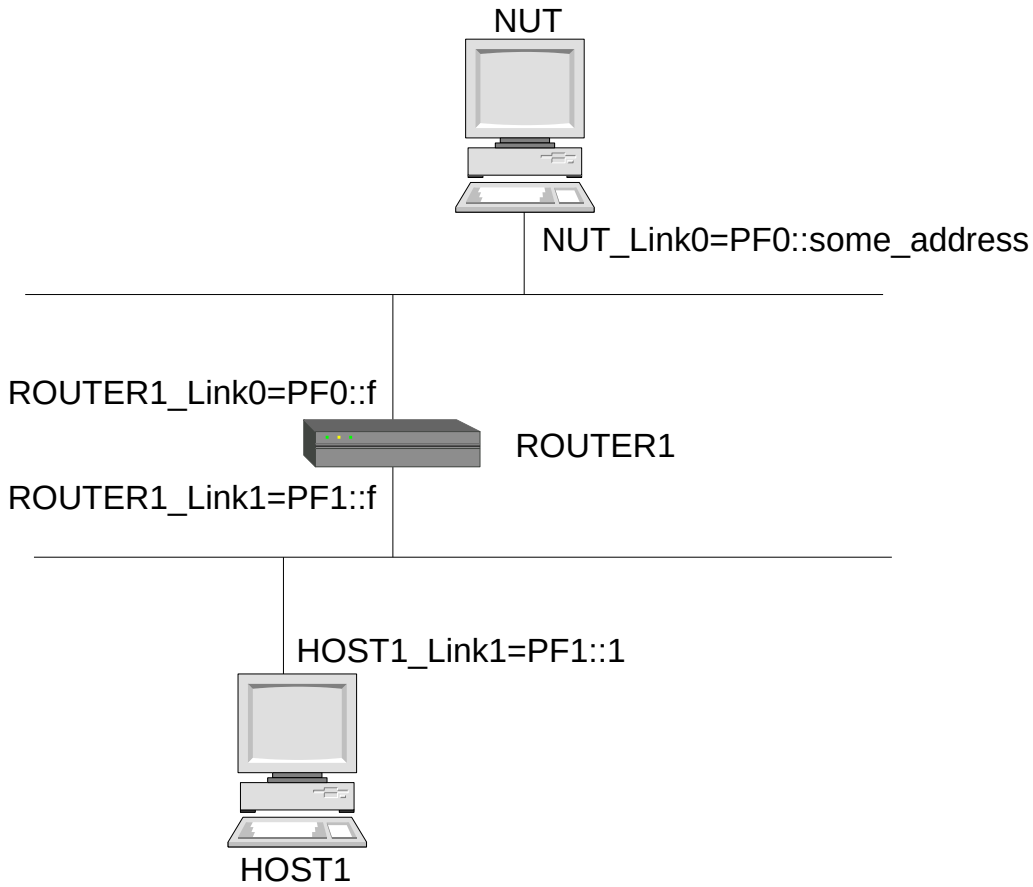
- Tunnel and Transport mode
- A combination of authentication and encryption algorithms
- Only ESP
- Manual key configuration

v6RL test suite coverage

- Tunnel and Transport mode
- A combination of authentication and encryption algorithms
- Only ESP
- Manual key configuration
- ICMPv6 messages exchange

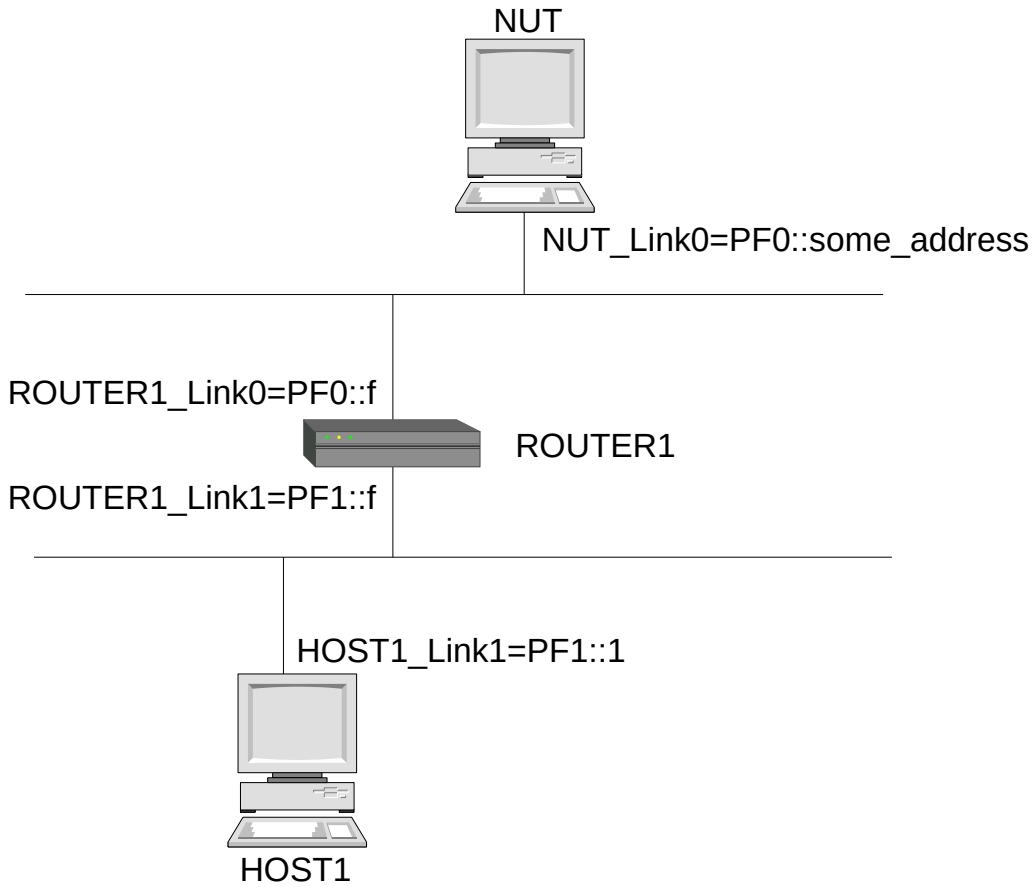
Test case 5.2.3

- Transport mode tested
- 3DES-CBC encryption algorithm
- NULL authentication algorithm



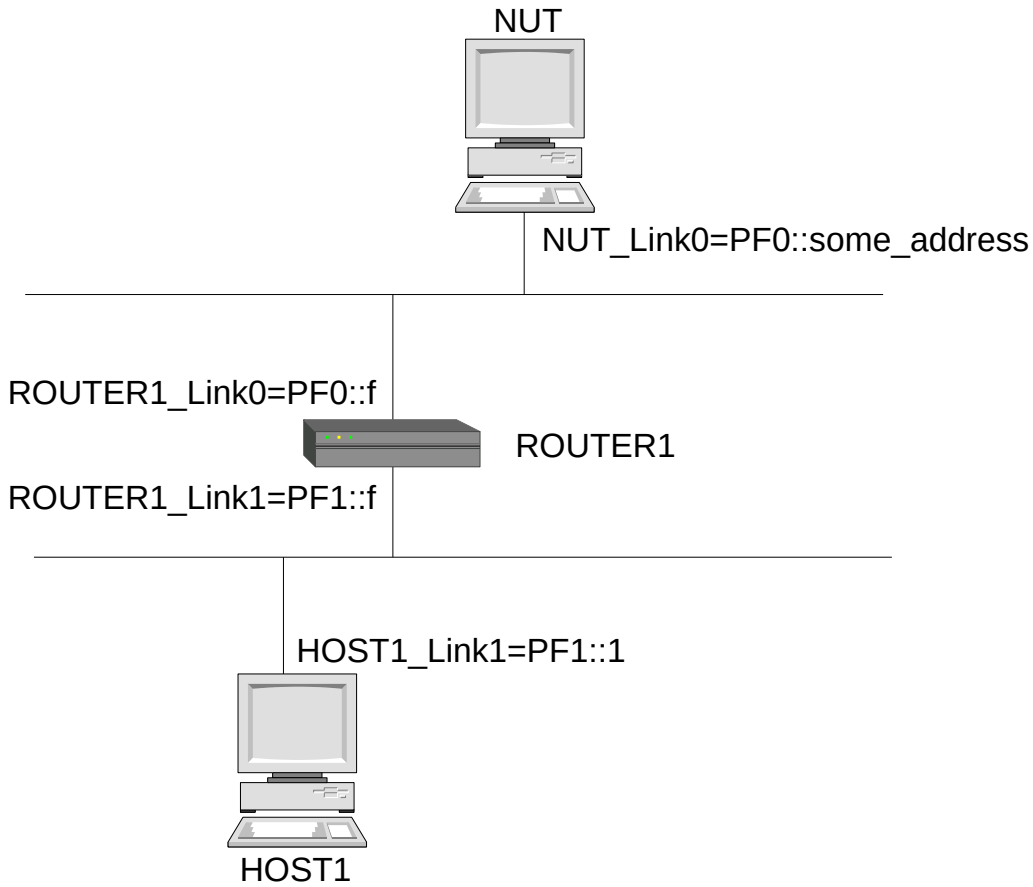
Test case 5.2.3

- Transport mode tested
- 3DES-CBC encryption algorithm
- NULL authentication algorithm



Test case 5.2.3

- Transport mode tested
- 3DES-CBC encryption algorithm
- NULL authentication algorithm



T3DevKit & IPv6 ATS

Why ?

- Helper tool for implementing TA-PA, TRI-SA and TCI-CD

T3DevKit & IPv6 ATS

Why ?

- Helper tool for implementing TA-PA, TRI-SA and TCI-CD
- Works in C++ environment, adequate for IPsec testing

T3DevKit & IPv6 ATS

Why ?

- Helper tool for implementing TA-PA, TRI-SA and TCI-CD
- Works in C++ environment, adequate for IPsec testing
- Existing IPv6 ATS enables code reuse (IPv6, ICMPv6, etc.)

T3DevKit & IPv6 ATS

Why ?

- Helper tool for implementing TA-PA, TRI-SA and TCI-CD
- Works in C++ environment, adequate for IPsec testing
- Existing IPv6 ATS enables code reuse (IPv6, ICMPv6, etc.)
- Freely available under CeCILL-C license

GNU crypto library

- General purpose cryptographic library

GNU crypto library

- General purpose cryptographic library
- Several cryptographic algorithms provided

GNU crypto library

- General purpose cryptographic library
- Several cryptographic algorithms provided
- All IPsec cryptographic functions implemented

GNU crypto library

- General purpose cryptographic library
- Several cryptographic algorithms provided
- All IPsec cryptographic functions implemented
- Broad user base and examples on-line

GNU crypto library

- General purpose cryptographic library
- Several cryptographic algorithms provided
- All IPsec cryptographic functions implemented
- Broad user base and examples on-line
- Freely available under LGPL license

Test case engineering

- Just an ICMPv6 Echo Request and Echo Reply exchanged
- Simple message sequence
- Messages use 3DES-CBC encryption with PSK
- Complex assembly and disassembly
- Where to perform cryptographic operations?

Test case engineering

- Just an ICMPv6 Echo Request and Echo Reply exchanged
- Simple message sequence
- Messages use 3DES-CBC encryption with PSK
- Complex assembly and disassembly
- Where to perform cryptographic operations?

Test case engineering

- Just an ICMPv6 Echo Request and Echo Reply exchanged
- Simple message sequence
- Messages use 3DES-CBC encryption with PSK
- Complex assembly and disassembly
- Where to perform cryptographic operations?

Test case engineering

- Just an ICMPv6 Echo Request and Echo Reply exchanged
- Simple message sequence
- Messages use 3DES-CBC encryption with PSK
- Complex assembly and disassembly
- Where to perform cryptographic operations?
  - CoDec
  - External Functions

CoDec only Transmission

- ESP message modeled in TTCN-3
- Checksum and padding fields calculated in the CoDec
- Payload encrypted in the CoDec

```
Link1.send(ICMPv6WithESP_EchoRequest_AuthNULL(SPI_SA1, ''O));
```

CoDec only Transmission

- ESP message modeled in TTCN-3
- Checksum and padding fields calculated in the CoDec
- Payload encrypted in the CoDec

```
Link1.send(ICMPv6WithESP_EchoRequest_AuthNULL(SPI_SA1, ''O));
```

CoDec only Transmission

- ESP message modeled in TTCN-3
- Checksum and padding fields calculated in the CoDec
- Payload encrypted in the CoDec

```
Link1.send(ICMPv6WithESP_EchoRequest_AuthNULL(SPI_SA1, ''O));
```

CoDec only Reception

```
alt
  //Receive the correct answer
  [] Link1.receive(ICMPv6WithESP_EchoReply_AuthNULL
                                        (SPI_SA2, ''O))
       { setverdict(pass);
         replyTimer.stop; }
  //Receive incorrect answer
  [] Link1.receive
       { setverdict(fail);
         replyTimer.stop; }
  //Receive no answer
  [] replyTimer.timeout
       { setverdict(fail); }
```

CoDec only Reception

```
alt
  //Receive the correct answer
  [] Link1.receive(ICMPv6WithESP_EchoReply_AuthNULL
                                        (SPI_SA2, ''O))
       { setverdict(pass);
         replyTimer.stop; }
  //Receive incorrect answer
  [] Link1.receive
       { setverdict(fail);
         replyTimer.stop; }
  //Receive no answer
  [] replyTimer.timeout
       { setverdict(fail); }
```

CoDec only Reception

```
alt
  //Receive the correct answer
  [] Link1.receive(ICMPv6WithESP_EchoReply_AuthNULL
                                        (SPI_SA2, ''O))
       { setverdict(pass);
         replyTimer.stop; }
  //Receive incorrect answer
  [] Link1.receive
       { setverdict(fail);
         replyTimer.stop; }
  //Receive no answer
  [] replyTimer.timeout
       { setverdict(fail); }
```

CoDec+Ext Transmission

```
template ESPMessage ICMPv6ESPMessage (IPv6AddressType src,
                    IPv6AddressType dst, octetstring m_spi,
                    octetstring m_data, UInt16 checksum) := {

          SPI:= m_spi,
          SeqNum := 1,
          Payload := EncryptPayload(src, dst, EchoRequestType,
                                    m_data, checksum),
          ICV :=omit
 }
```

CoDec+Ext Reception

```
alt{
        //Receive correct answer, unverified encrypted payload
        [] Link1.receive(ICMPv6ESPMessage_Answer_AuthNULL
                        (PF0_1, PF1_1, SPI_SA2, DATA, checksum)) -> value Myvar {
              var bitstring encpayload := Myvar.Payload;
              var UInt8 payloadLength := lengthof(encpayload)/8;
              var EncPayload payload := DecriptPayload(encpayload, payloadLength);
              if (match(payload, ICMPv6EncPayload_Answer(PF0_1, PF1_1, DATA))) {
                     setverdict(pass);
              } else {
                     setverdict(fail);
              }
              replyTimer.stop;
        }
        //Receive incorrect answer
        [] Link1.receive {
              setverdict(fail);
              replyTimer.stop;
        }
        //Receive no answer
        [] replyTimer.timeout {
              setverdict(fail);
        }
  }
```

CoDec+Ext Reception

```
alt{
        //Receive correct answer, unverified encrypted payload
        [] Link1.receive(ICMPv6ESPMessage_Answer_AuthNULL
                        (PF0_1, PF1_1, SPI_SA2, DATA, checksum)) -> value Myvar {
            var bitstring encpayload := Myvar.Payload;
            var UInt8 payloadLength := lengthof(encpayload)/8;
            var EncPayload payload := DecriptPayload(encpayload, payloadLength);
            if (match(payload, ICMPv6EncPayload_Answer(PF0_1, PF1_1, DATA))) {
                setverdict(pass);
            } else {
                setverdict(fail);
            }
            replyTimer.stop;
        }
        //Receive incorrect answer
        [] Link1.receive {
            setverdict(fail);
            replyTimer.stop;
        }
        //Receive no answer
        [] replyTimer.timeout {
            setverdict(fail);
        }
}
```

CoDec+Ext Reception

```
alt{
        //Receive correct answer, unverified encrypted payload
        [] Link1.receive(ICMPv6ESPMessage_Answer_AuthNULL
                        (PF0_1, PF1_1, SPI_SA2, DATA, checksum)) -> value Myvar {
            var bitstring encpayload := Myvar.Payload;
            var UInt8 payloadLength := lengthof(encpayload)/8;
            var EncPayload payload := DecriptPayload(encpayload, payloadLength);
            if (match(payload, ICMPv6EncPayload_Answer(PF0_1, PF1_1, DATA))) {
                setverdict(pass);
            } else {
                setverdict(fail);
            }
            replyTimer.stop;
        }
        //Receive incorrect answer
        [] Link1.receive {
            setverdict(fail);
            replyTimer.stop;
        }
        //Receive no answer
        [] replyTimer.timeout {
            setverdict(fail);
        }
}
```

Message transmission & reception

CoDec

- High ATS abstraction (too much?)

External Functions

Message transmission & reception

CoDec

- High ATS abstraction (too much?)

External Functions

- More control from ATS

Message transmission & reception

CoDec

- High ATS abstraction (too much?)
- Increased CoDec complexity

External Functions

- More control from ATS

Message transmission & reception

CoDec

- High ATS abstraction (too much?)
- Increased CoDec complexity

External Functions

- More control from ATS
- CoDec just encode and decode

Message transmission & reception

CoDec

- High ATS abstraction (too much?)
- Increased CoDec complexity
- Difficult code factorization and reuse

External Functions

- More control from ATS
- CoDec just encode and decode

Message transmission & reception

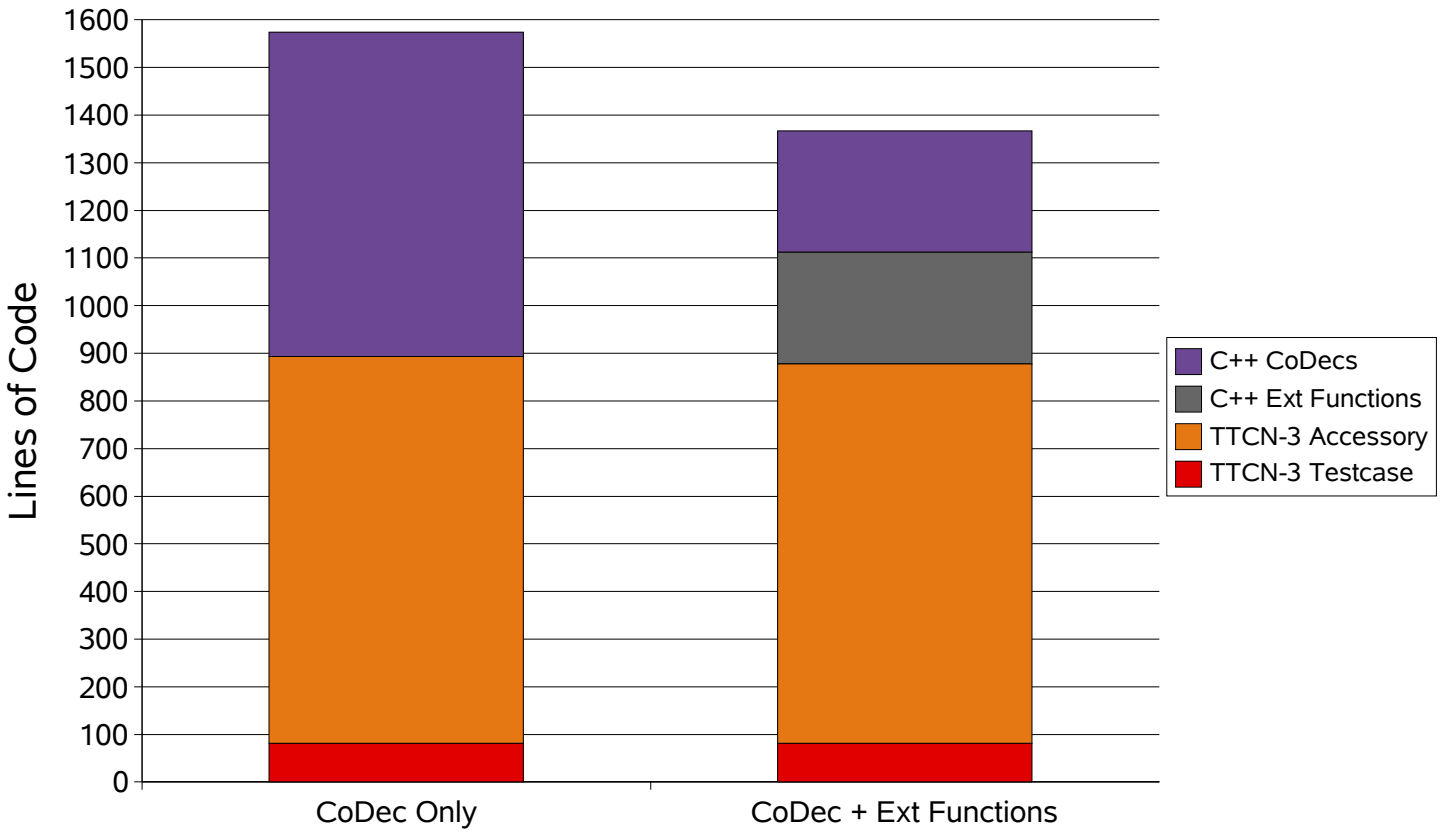
CoDec

- High ATS abstraction (too much?)
- Increased CoDec complexity
- Difficult code factorization and reuse

External Functions

- More control from ATS
- CoDec just encode and decode
- Software engineering techniques applicable

loc based metrics



Performance

- Every time an external function is invoked, encode and decode operations on the CoDec are invoked

Performance

- Every time an external function is invoked, encode and decode operations on the CoDec are invoked
- External functions based approach requires 4 external function invocations.

Performance

- Every time an external function is invoked, encode and decode operations on the CoDec are invoked
- External functions based approach requires 4 external function invocations.
- Not relevant in conformance or interoperability testing, but might be critical for other test paradigms.

Final remarks

- Ongoing research for more thorough analysis
- Both methodologies are valid and applicable, with consistent results
- Excessively complex CoDec development diverges from TTCN-3 philosophy
- When performance degradation is allowed, external functions provide better code properties and a cleaner solution

Final remarks

- Ongoing research for more thorough analysis
- Both methodologies are valid and applicable, with consistent results
- Excessively complex CoDec development diverges from TTCN-3 philosophy
- When performance degradation is allowed, external functions provide better code properties and a cleaner solution

Final remarks

- Ongoing research for more thorough analysis
- Both methodologies are valid and applicable, with consistent results
- Excessively complex CoDec development diverges from TTCN-3 philosophy
- When performance degradation is allowed, external functions provide better code properties and a cleaner solution

Final remarks

- Ongoing research for more thorough analysis
- Both methodologies are valid and applicable, with consistent results
- Excessively complex CoDec development diverges from TTCN-3 philosophy
- When performance degradation is allowed, external functions provide better code properties and a cleaner solution

Thank you for your time

Questions?

http://www.irisa.fr/tipi/publi/t3uc2007paper.pdf