SPONSORED BY THE

Federal Ministry of Education and Research

TLS-Anvil

Adapting Combinatorial Testing for TLS Libraries

RWC 2023

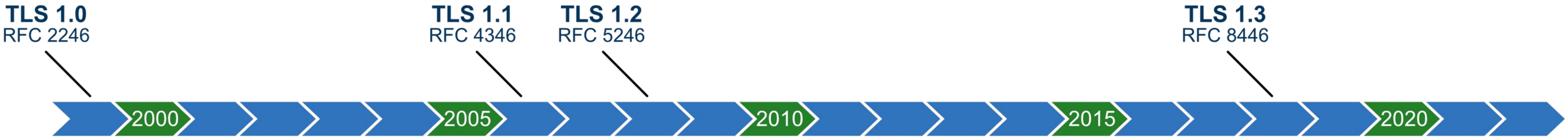
Marcel Maehren[1], Philipp Nieting[1], Sven Hebrok[2], Robert Merget[3], Juraj Somorovsky[2], Jörg Schwenk[1]

1 Ruhr University Bochum

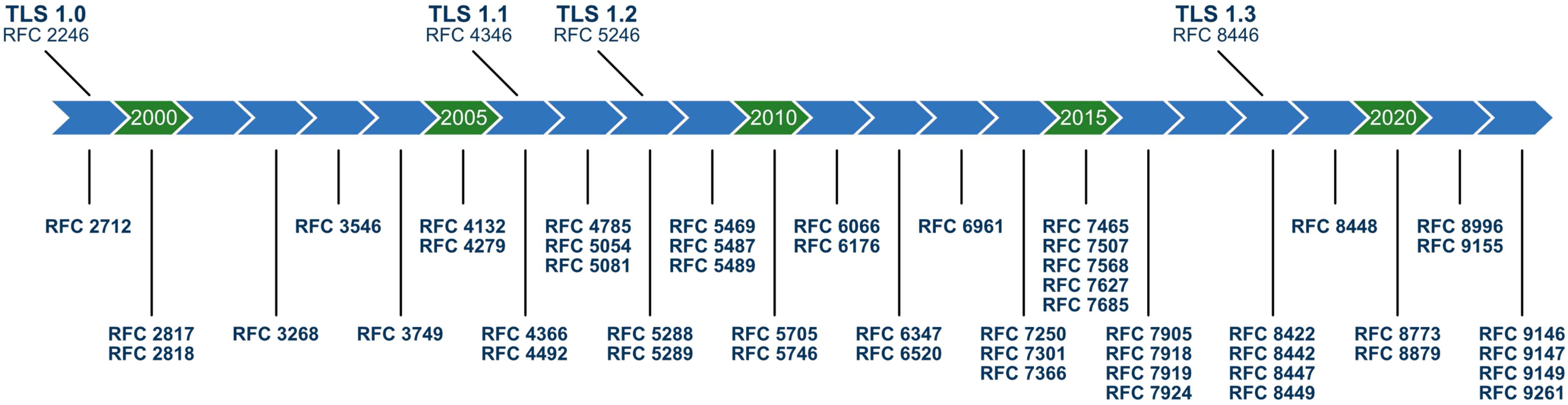
2 Paderborn University

3 Technology Innovation Institute

TLS Is a Complex Protocol



TLS Is a Complex Protocol



TLS 1.0
RFC 2246

TLS 1.1
RFC 4346

TLS 1.2
RFC 5246

TLS 1.3
RFC 8446

2000

2005

2010

2015

2020

RFC 2712

RFC 2817
RFC 2818

RFC 3268

RFC 3546

RFC 3749

RFC 4132
RFC 4279

RFC 4366
RFC 4492

RFC 4785
RFC 5054
RFC 5081

RFC 5288
RFC 5289

RFC 5469
RFC 5487
RFC 5489

RFC 5705
RFC 5746

RFC 6066
RFC 6176

RFC 6347
RFC 6520

RFC 6961

RFC 7250
RFC 7301
RFC 7366

RFC 7465
RFC 7507
RFC 7568
RFC 7627
RFC 7685

RFC 7905
RFC 7918
RFC 7919
RFC 7924

RFC 8422
RFC 8442
RFC 8447
RFC 8449

RFC 8448

RFC 8773
RFC 8879

RFC 8996
RFC 9155

RFC 9146
RFC 9147
RFC 9149
RFC 9261

RFC Requirement Example

> The receiver **MUST** check [the] padding and **MUST** use the **bad_record_mac alert** to indicate padding errors.
>
> - RFC 5246 (CBC Block Cipher)

Security measure to avoid **Padding Oracle** attacks

→ Requirement must be met regardless of negotiated parameters

Parameters Example

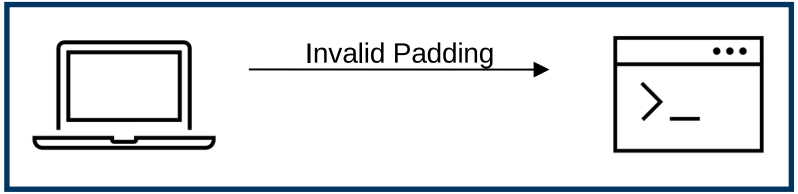
DHE

RSA

AES

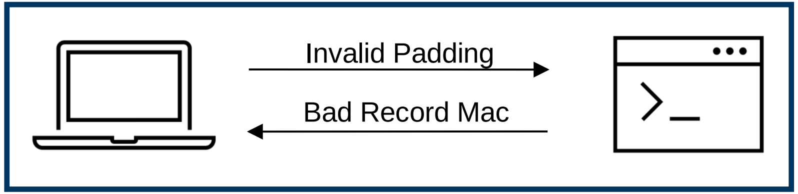
Parameters Example

DHE RSA AES



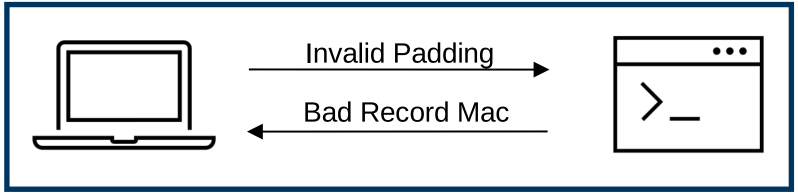
Parameters Example

DHE RSA AES



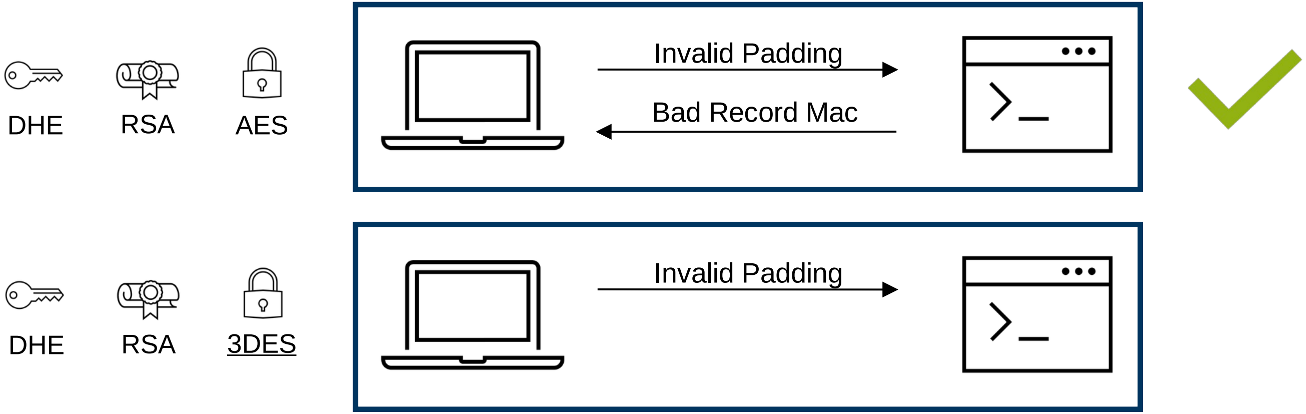
Parameters Example

DHE RSA AES



✓

Parameters Example



DHE RSA AES

Invalid Padding

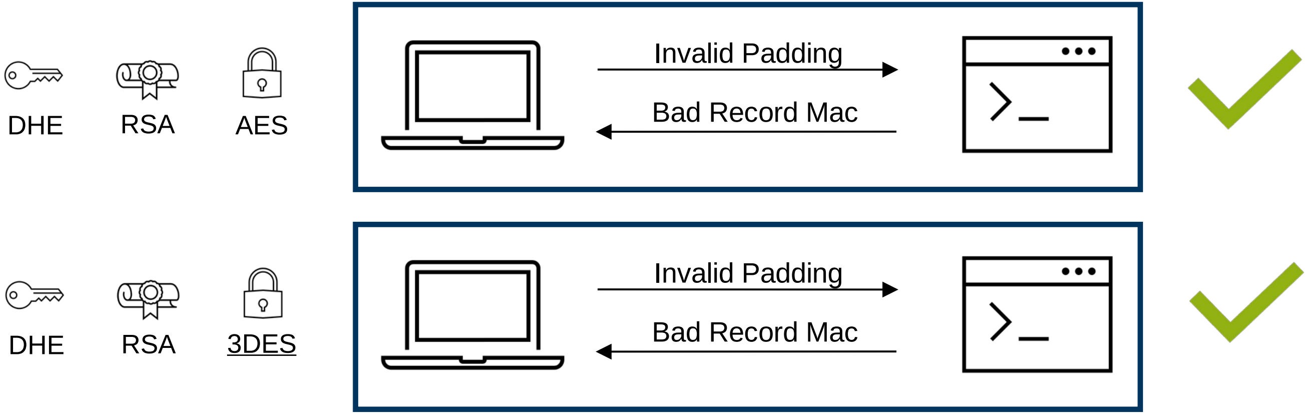
Bad Record Mac

✓

DHE RSA 3DES

Invalid Padding

Parameters Example



DHE RSA AES

Invalid Padding →

← Bad Record Mac

✓

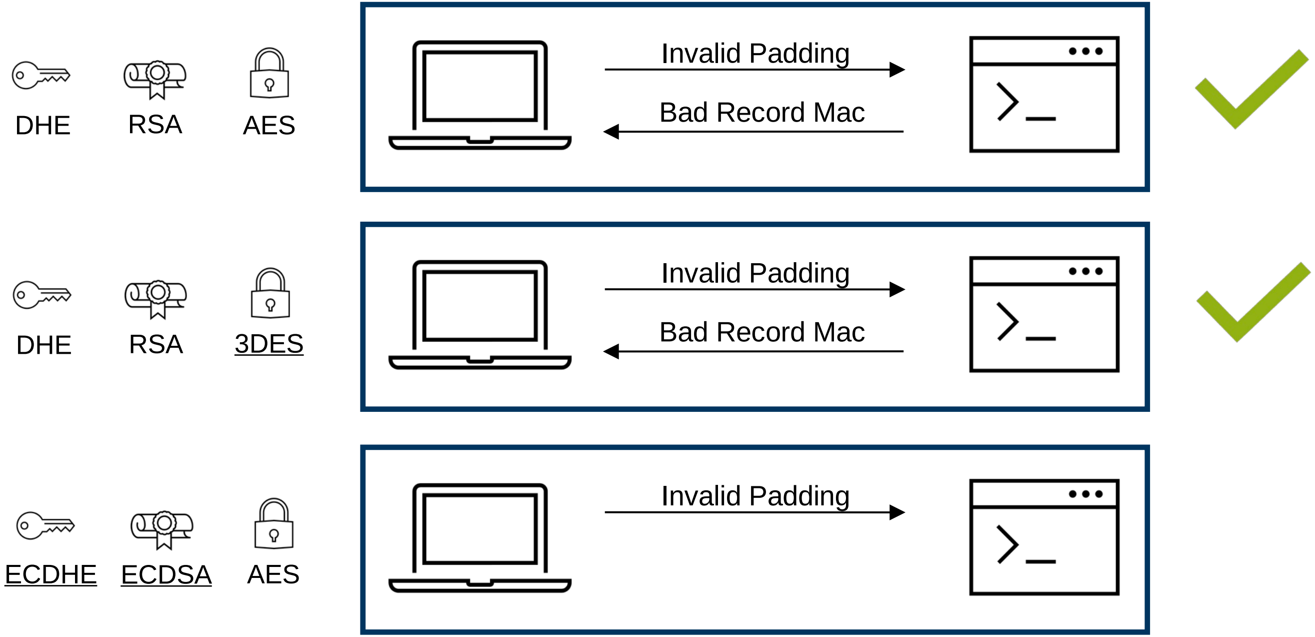
DHE RSA 3DES

Invalid Padding →

← Bad Record Mac

✓

Parameters Example



DHE RSA AES

Invalid Padding →

← Bad Record Mac

✓

DHE RSA 3DES

Invalid Padding →

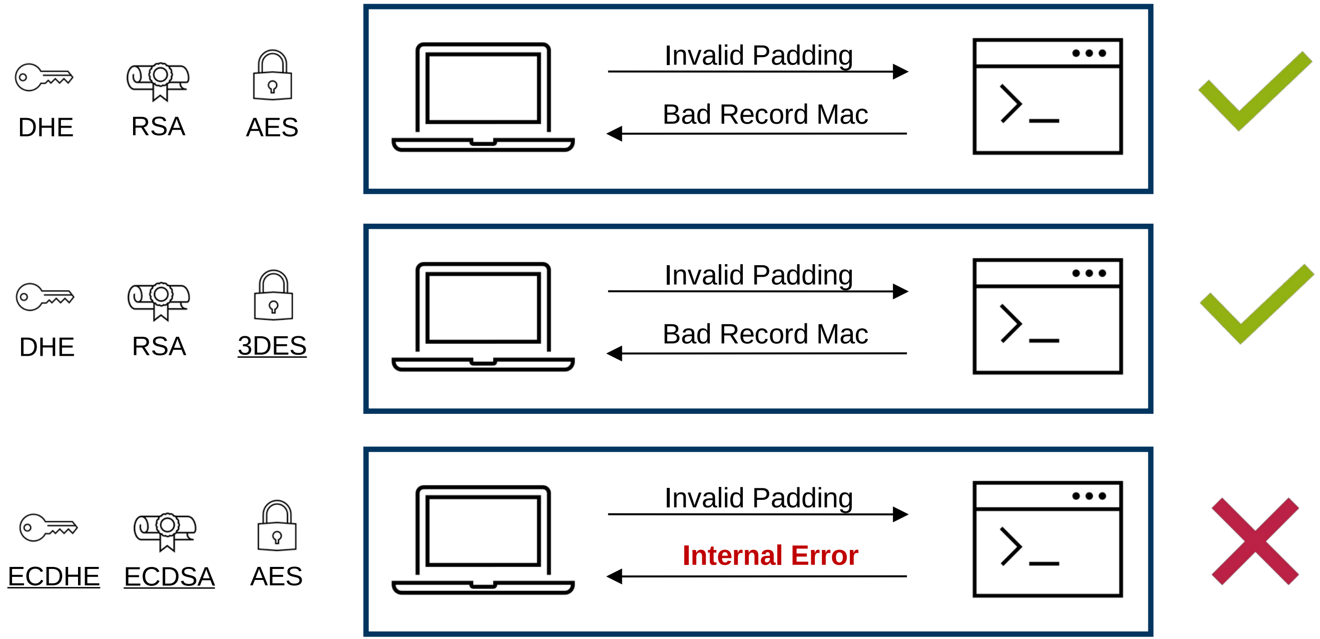
← Bad Record Mac

✓

ECDHE ECDSA AES

Invalid Padding →

Parameters Example



DHE RSA AES

Invalid Padding

Bad Record Mac

✓

DHE RSA 3DES

Invalid Padding

Bad Record Mac

✓

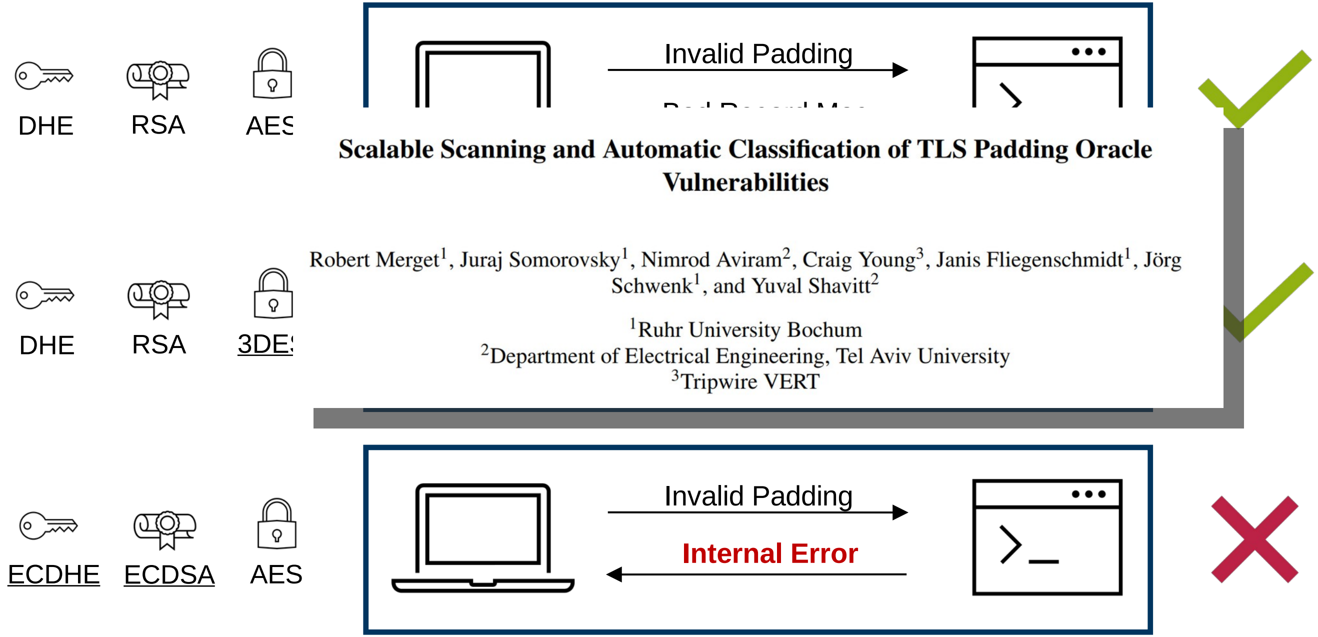
ECDHE ECDSA AES

Invalid Padding

Internal Error

✗

Parameters Example



Scalable Scanning and Automatic Classification of TLS Padding Oracle Vulnerabilities

Robert Merget[1], Juraj Somorovsky[1], Nimrod Aviram[2], Craig Young[3], Janis Fliegenschmidt[1], Jörg Schwenk[1], and Yuval Shavitt[2]

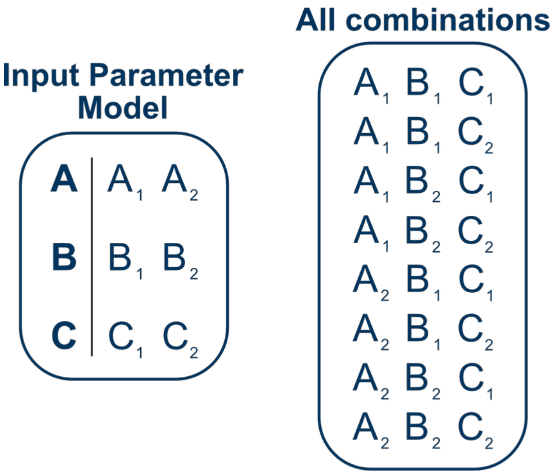
[1]Ruhr University Bochum
[2]Department of Electrical Engineering, Tel Aviv University
[3]Tripwire VERT

t-way Testing

Input Parameter Model

A	A_1	A_2
B	B_1	B_2
C	C_1	C_2

t-way Testing



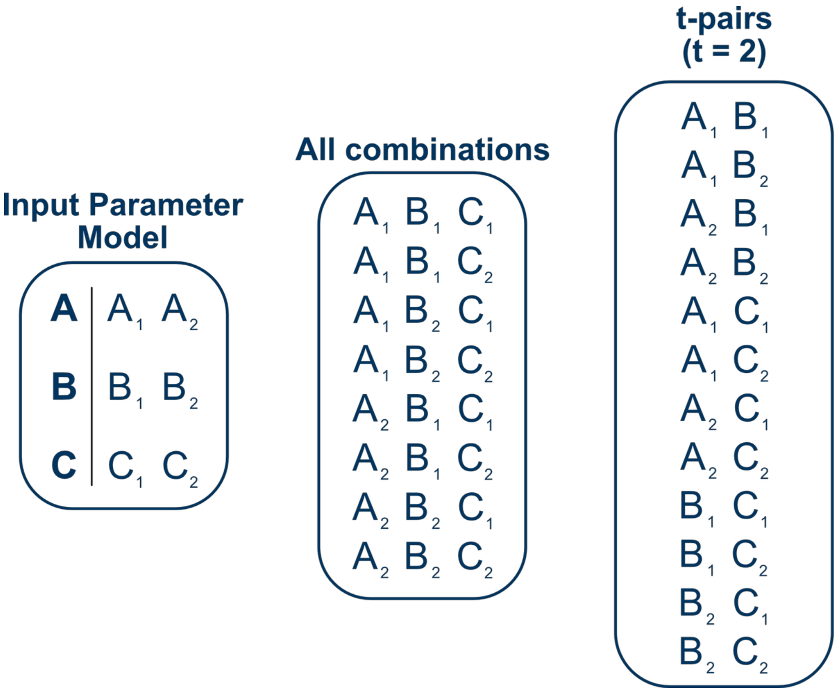
Input Parameter Model

A	A_1	A_2
B	B_1	B_2
C	C_1	C_2

All combinations

A_1	B_1	C_1
A_1	B_1	C_2
A_1	B_2	C_1
A_1	B_2	C_2
A_2	B_1	C_1
A_2	B_1	C_2
A_2	B_2	C_1
A_2	B_2	C_2

t-way Testing



Input Parameter Model

A	A_1	A_2
B	B_1	B_2
C	C_1	C_2

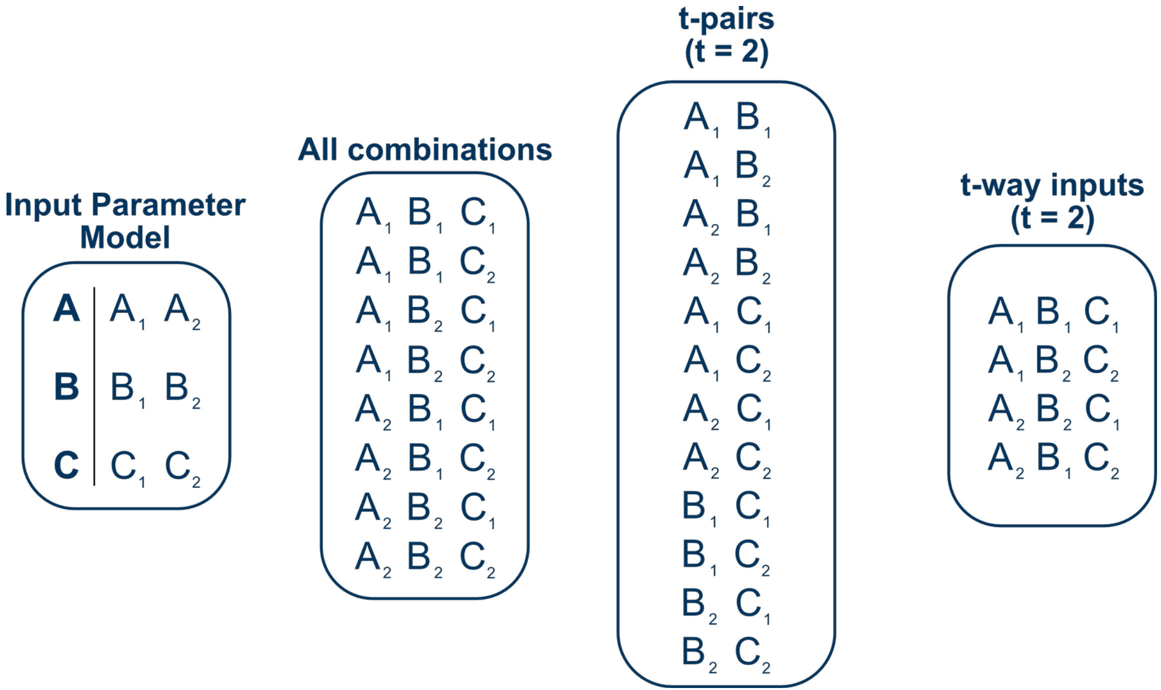
All combinations

$A_1\ B_1\ C_1$
$A_1\ B_1\ C_2$
$A_1\ B_2\ C_1$
$A_1\ B_2\ C_2$
$A_2\ B_1\ C_1$
$A_2\ B_1\ C_2$
$A_2\ B_2\ C_1$
$A_2\ B_2\ C_2$

t-pairs
(t = 2)

$A_1\ B_1$
$A_1\ B_2$
$A_2\ B_1$
$A_2\ B_2$
$A_1\ C_1$
$A_1\ C_2$
$A_2\ C_1$
$A_2\ C_2$
$B_1\ C_1$
$B_1\ C_2$
$B_2\ C_1$
$B_2\ C_2$

t-way Testing



Input Parameter Model

A	A_1	A_2
B	B_1	B_2
C	C_1	C_2

All combinations

A_1 B_1 C_1
A_1 B_1 C_2
A_1 B_2 C_1
A_1 B_2 C_2
A_2 B_1 C_1
A_2 B_1 C_2
A_2 B_2 C_1
A_2 B_2 C_2

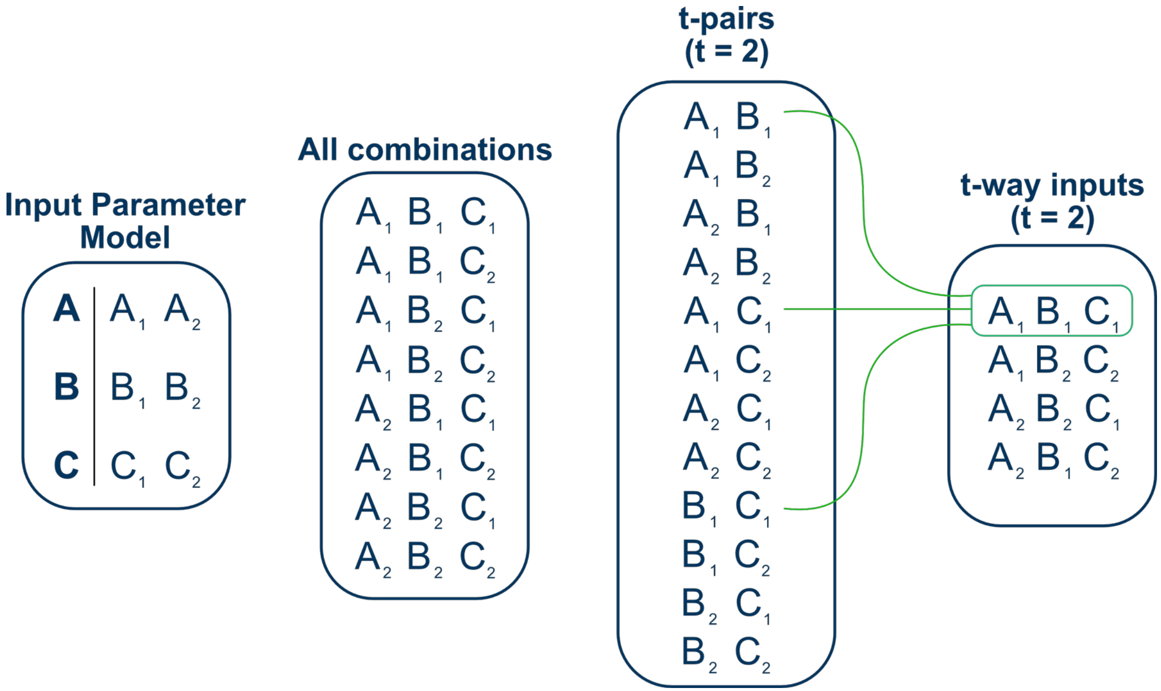
t-pairs (t = 2)

A_1 B_1
A_1 B_2
A_2 B_1
A_2 B_2
A_1 C_1
A_1 C_2
A_2 C_1
A_2 C_2
B_1 C_1
B_1 C_2
B_2 C_1
B_2 C_2

t-way inputs (t = 2)

A_1 B_1 C_1
A_1 B_2 C_2
A_2 B_2 C_1
A_2 B_1 C_2

t-way Testing



Input Parameter Model

A	A_1 A_2
B	B_1 B_2
C	C_1 C_2

All combinations

A_1 B_1 C_1
A_1 B_1 C_2
A_1 B_2 C_1
A_1 B_2 C_2
A_2 B_1 C_1
A_2 B_1 C_2
A_2 B_2 C_1
A_2 B_2 C_2

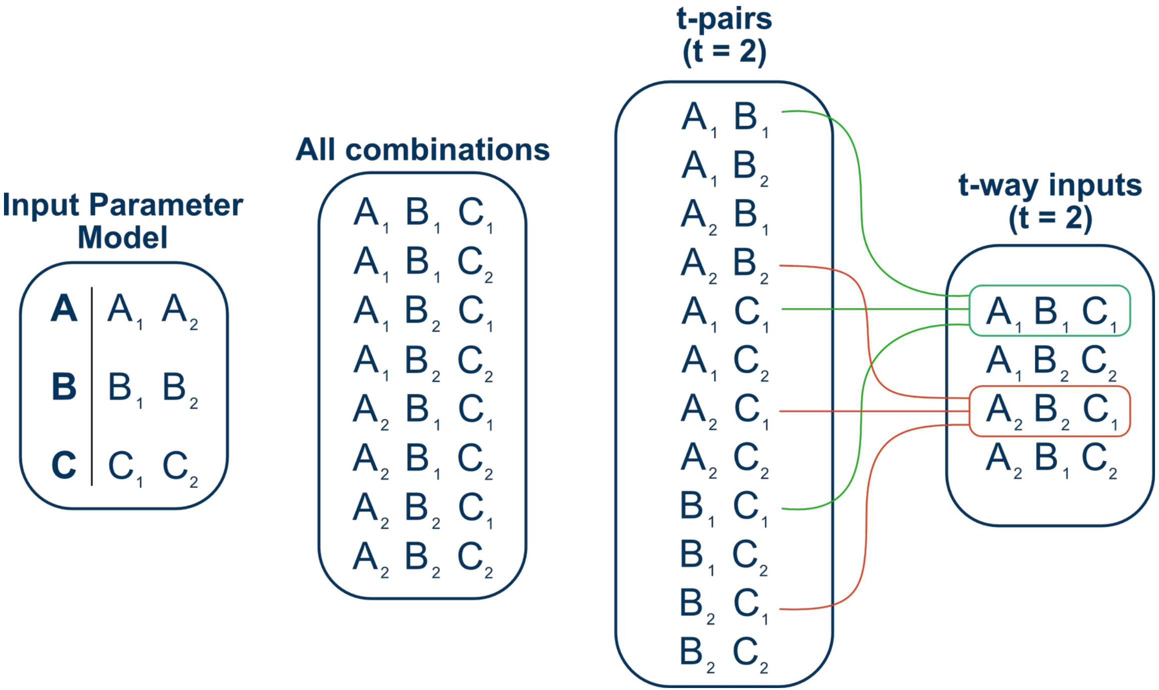
t-pairs
(t = 2)

A_1 B_1
A_1 B_2
A_2 B_1
A_2 B_2
A_1 C_1
A_1 C_2
A_2 C_1
A_2 C_2
B_1 C_1
B_1 C_2
B_2 C_1
B_2 C_2

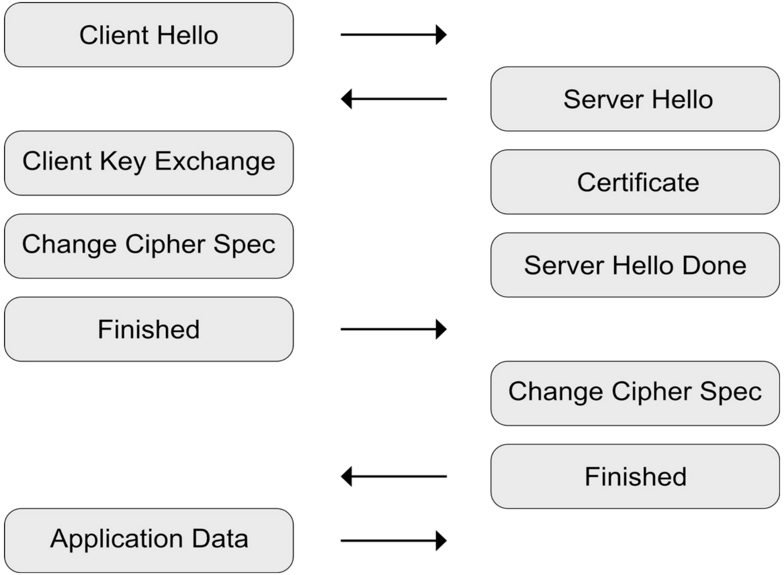
t-way inputs
(t = 2)

A_1 B_1 C_1
A_1 B_2 C_2
A_2 B_2 C_1
A_2 B_1 C_2

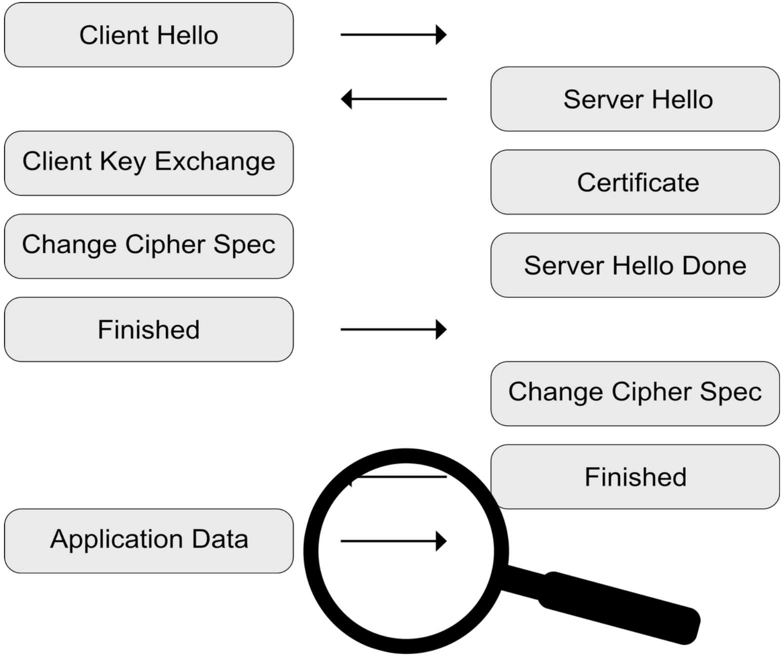
t-way Testing



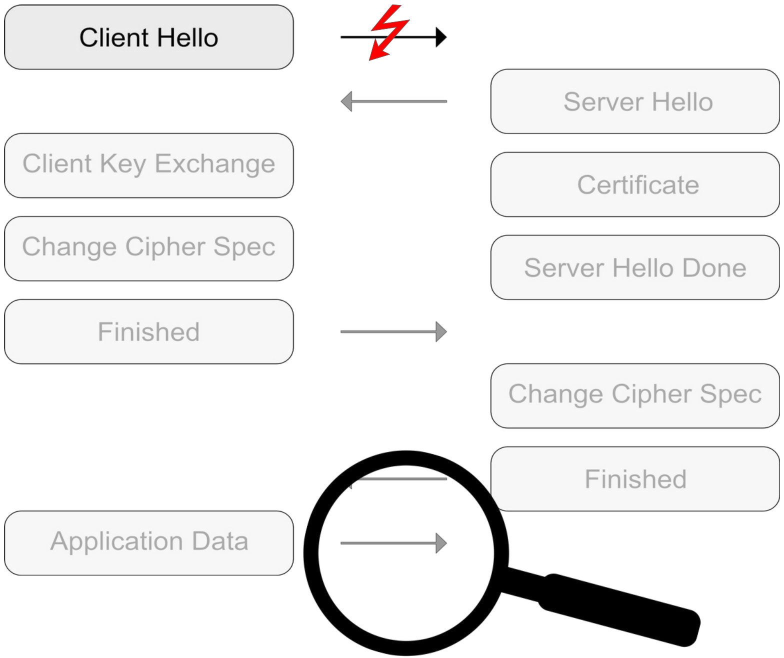
Test Inputs Must Be Constrained



Test Inputs Must Be Constrained



Test Inputs Must Be Constrained



Test Inputs Must Be Constrained

- Not all parameter values must be supported by a library

AES ✓ 3DES ✗ RSA 4096 ✓ RSA 2048 ✗

Test Inputs Must Be Constrained

- Not all parameter values must be supported by a library

AES ✓ 3DES ✗ RSA 4096 ✓ RSA 2048 ✗

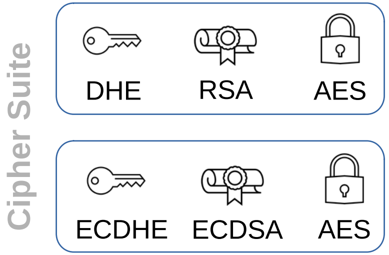
- Not all parameter values may be combined

Test Inputs Must Be Constrained

- Not all parameter values must be supported by a library

AES ✓ 3DES ✗ RSA 4096 ✓ RSA 2048 ✗

- Not all parameter values may be combined

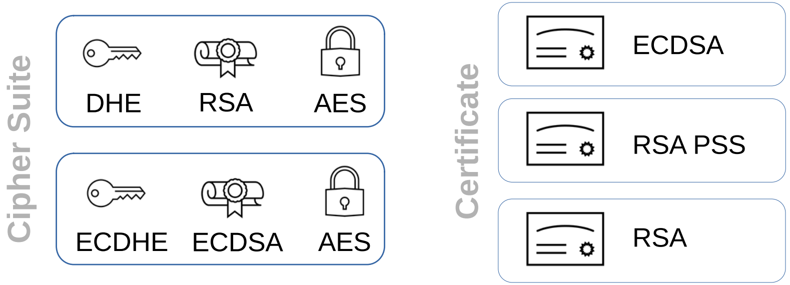


Test Inputs Must Be Constrained

- Not all parameter values must be supported by a library

AES ✓ 3DES ✗ RSA 4096 ✓ RSA 2048 ✗

- Not all parameter values may be combined

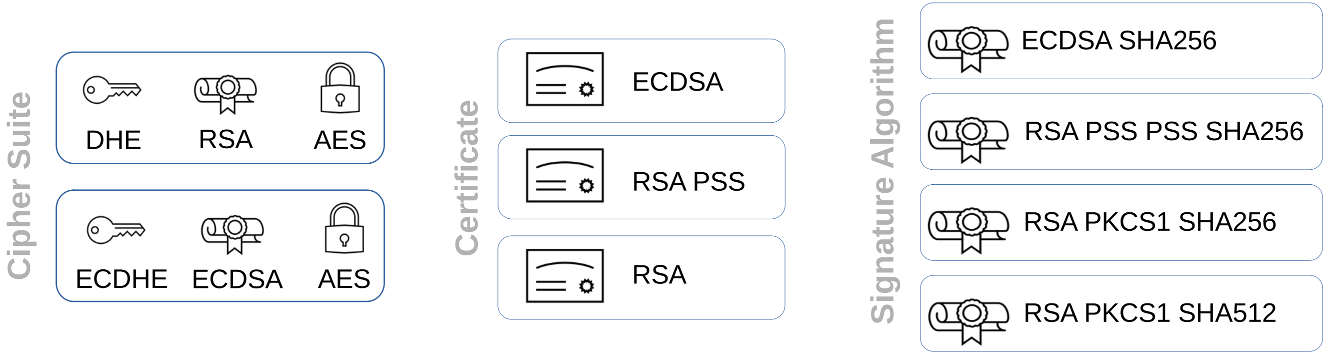


Test Inputs Must Be Constrained

- Not all parameter values must be supported by a library

AES ✓ 3DES ✗ RSA 4096 ✓ RSA 2048 ✗

- Not all parameter values may be combined



Cipher Suite

DHE RSA AES

ECDHE ECDSA AES

Certificate

ECDSA

RSA PSS

RSA

Signature Algorithm

ECDSA SHA256

RSA PSS PSS SHA256

RSA PKCS1 SHA256

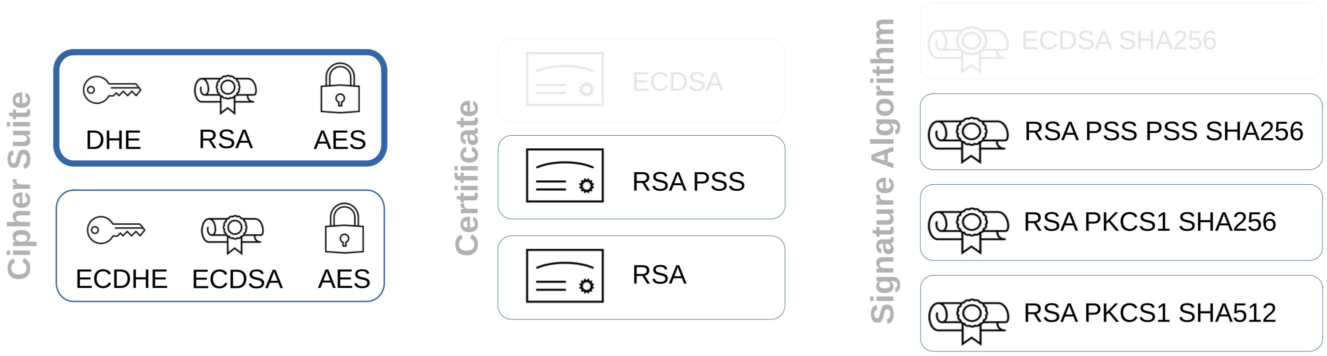
RSA PKCS1 SHA512

Test Inputs Must Be Constrained

- Not all parameter values must be supported by a library

AES ✓ 3DES ✗ RSA 4096 ✓ RSA 2048 ✗

- Not all parameter values may be combined



Cipher Suite

DHE RSA AES

ECDHE ECDSA AES

Certificate

ECDSA

RSA PSS

RSA

Signature Algorithm

ECDSA SHA256

RSA PSS PSS SHA256

RSA PKCS1 SHA256

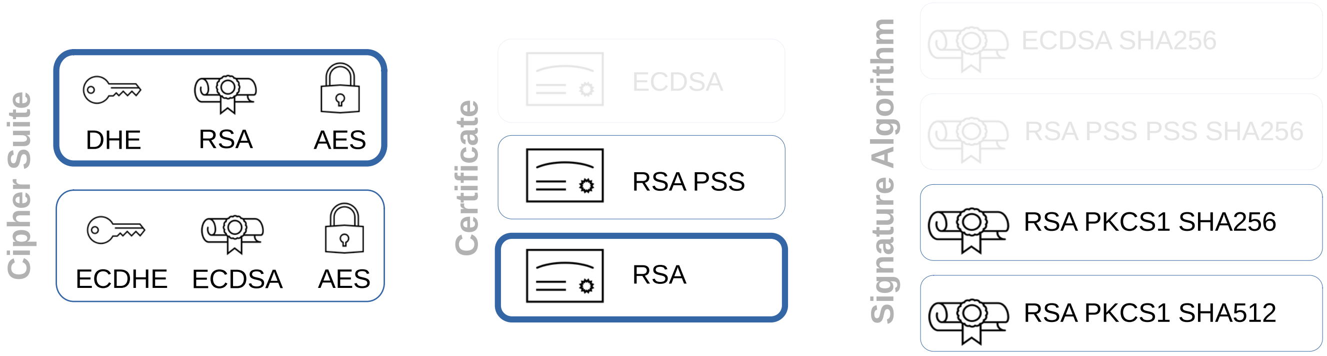
RSA PKCS1 SHA512

Test Inputs Must Be Constrained

- Not all parameter values must be supported by a library

AES ✓ 3DES ✗ RSA 4096 ✓ RSA 2048 ✗

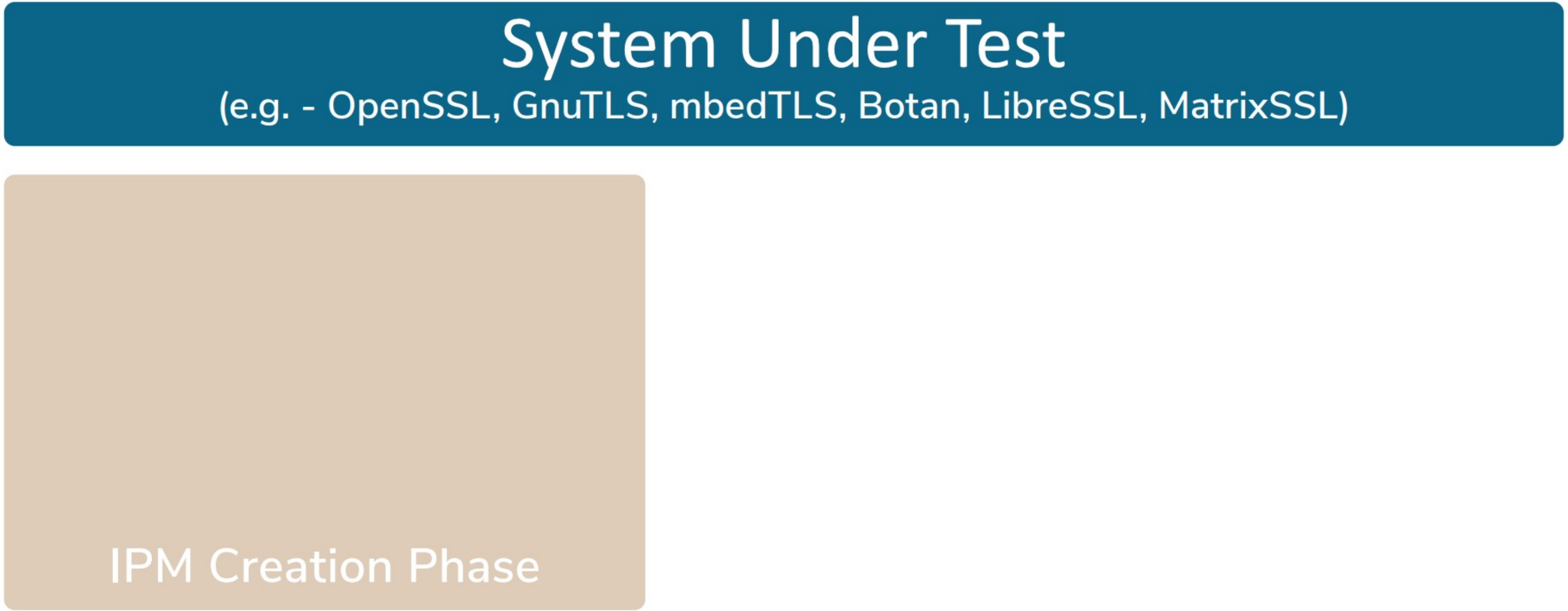
- Not all parameter values may be combined



TLS-Anvil

- TLS test suite for **black box** evaluation of clients and servers
- **t-way coverage** of parameters with carefully constrained inputs
- Based on **mandatory** RFC statements
- Up to 14 parameters considered
- 408 test templates based on 13 TLS RFCs

Execution

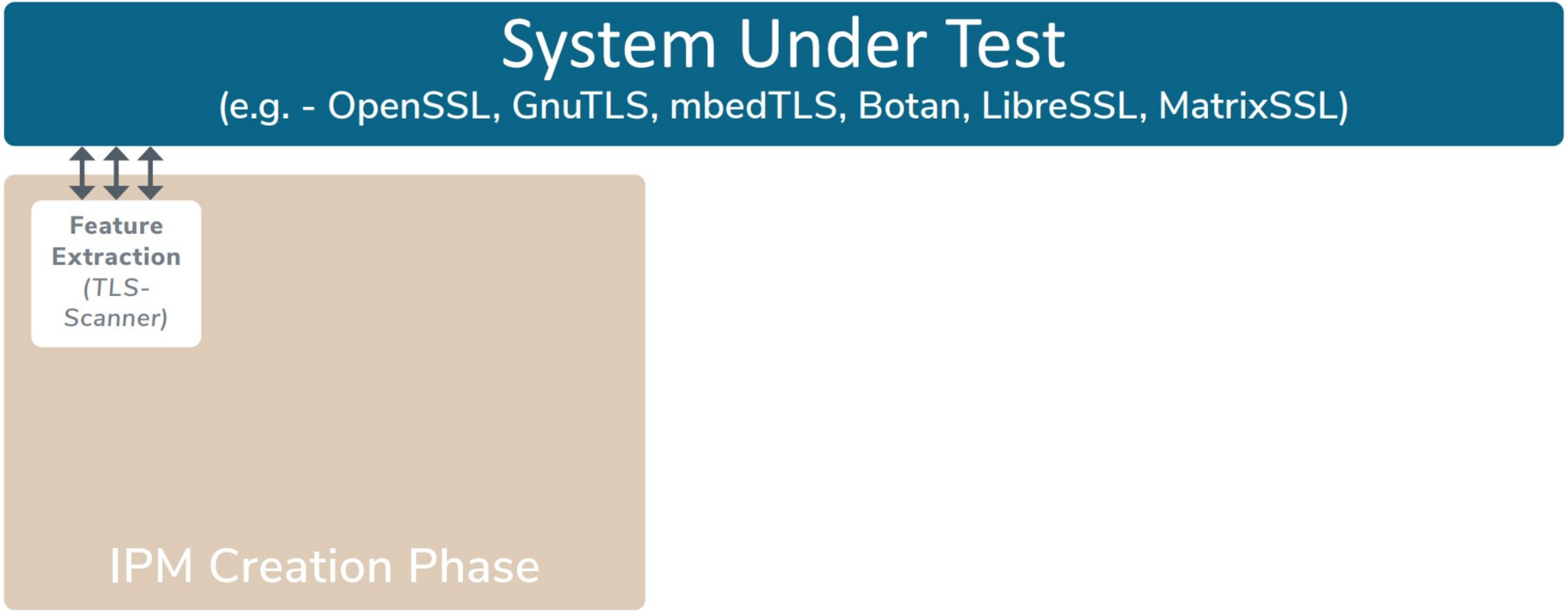


System Under Test

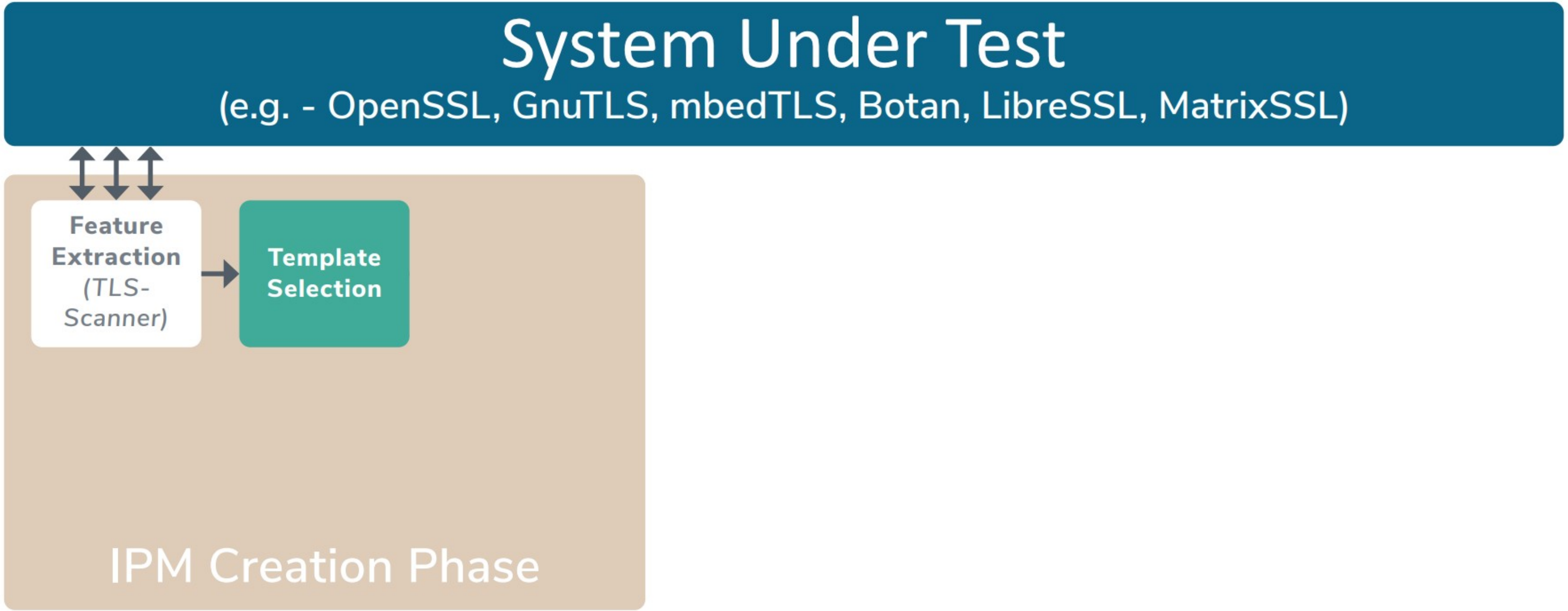
(e.g. - OpenSSL, GnuTLS, mbedTLS, Botan, LibreSSL, MatrixSSL)

IPM Creation Phase

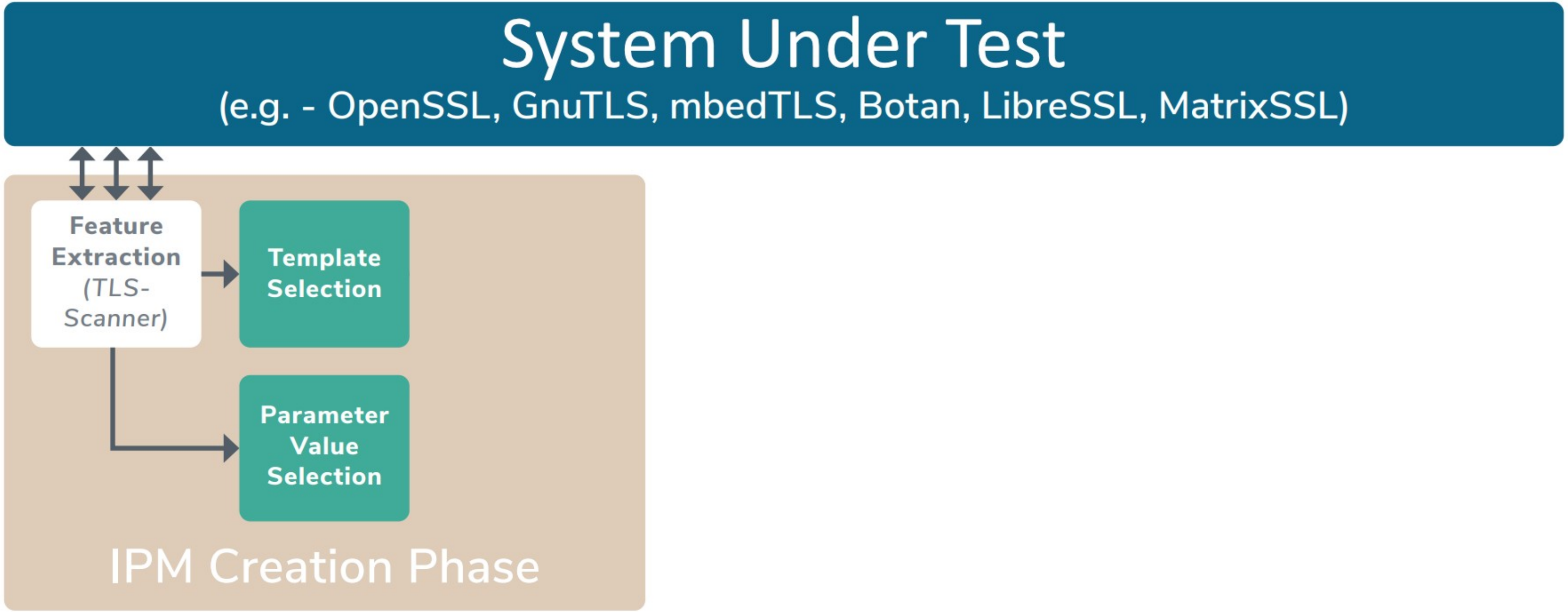
Execution


System Under Test
(e.g. - OpenSSL, GnuTLS, mbedTLS, Botan, LibreSSL, MatrixSSL)
Feature Extraction (TLS-Scanner)
IPM Creation Phase

Execution


System Under Test
(e.g. - OpenSSL, GnuTLS, mbedTLS, Botan, LibreSSL, MatrixSSL)
Feature Extraction (TLS-Scanner)
Template Selection
IPM Creation Phase

Execution



System Under Test

(e.g. - OpenSSL, GnuTLS, mbedTLS, Botan, LibreSSL, MatrixSSL)

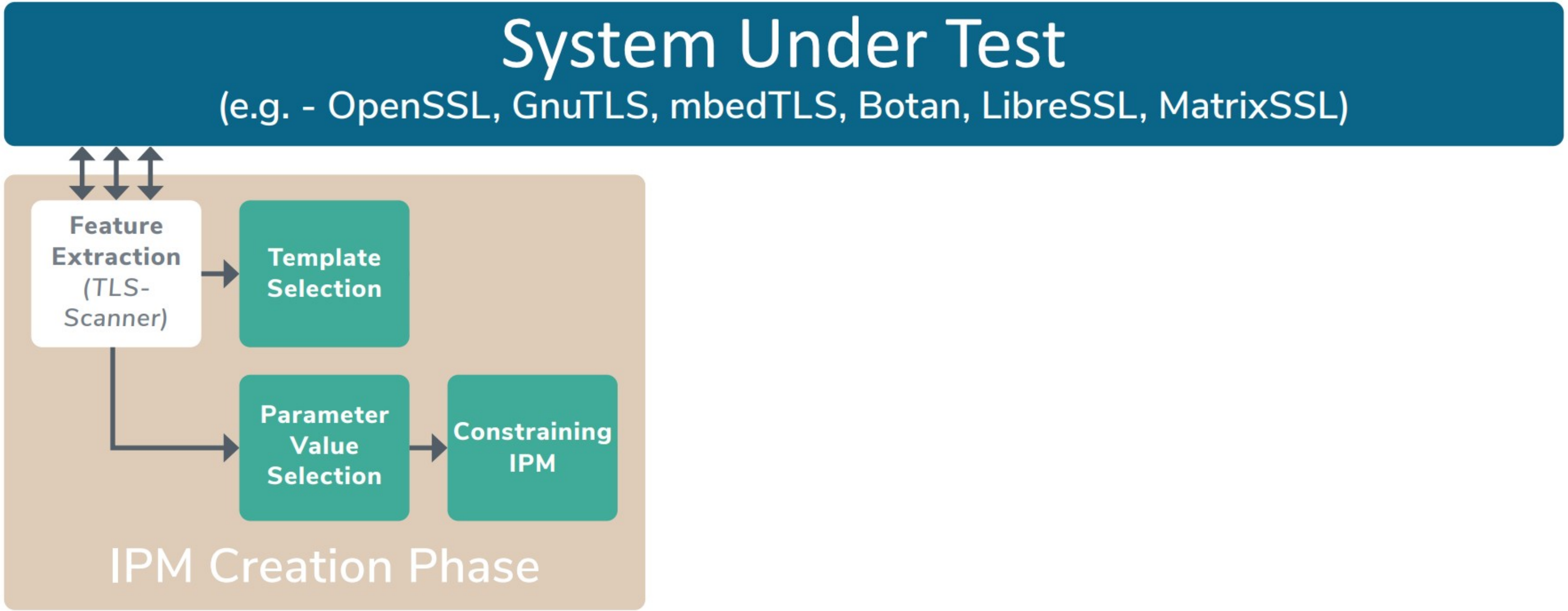
Feature Extraction (TLS-Scanner)

Template Selection

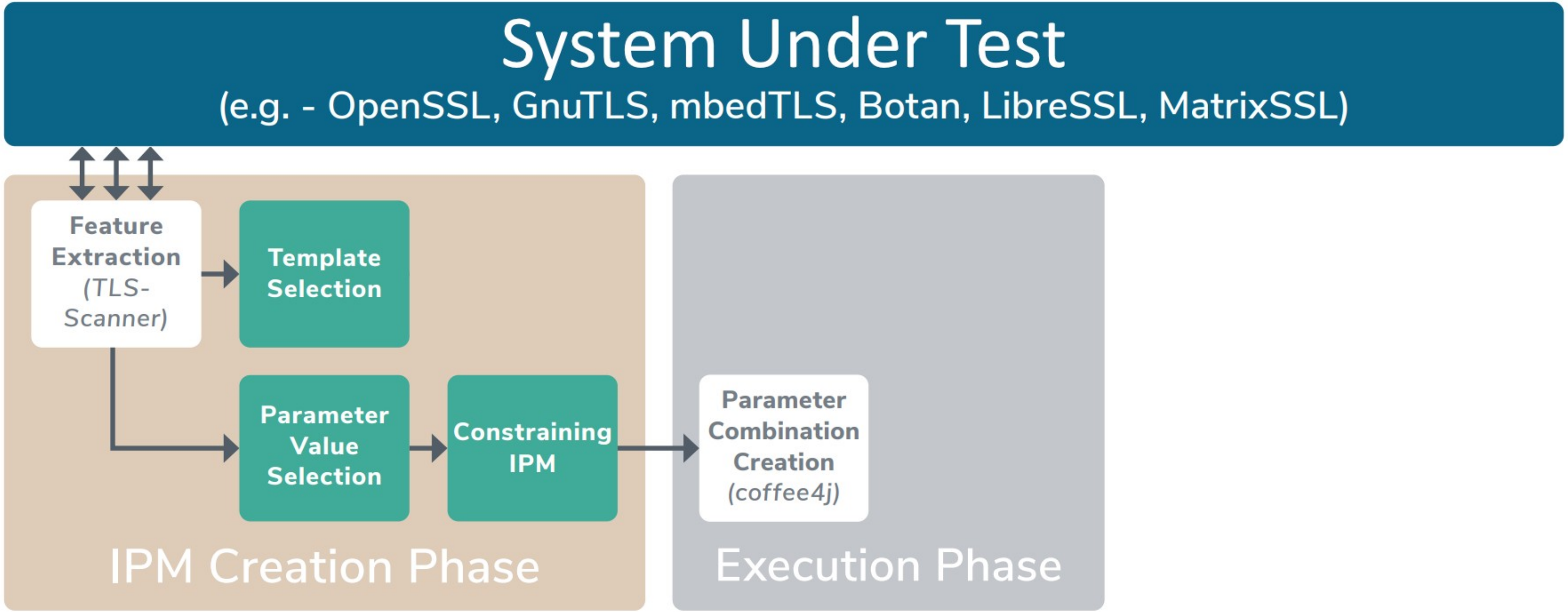
Parameter Value Selection

IPM Creation Phase

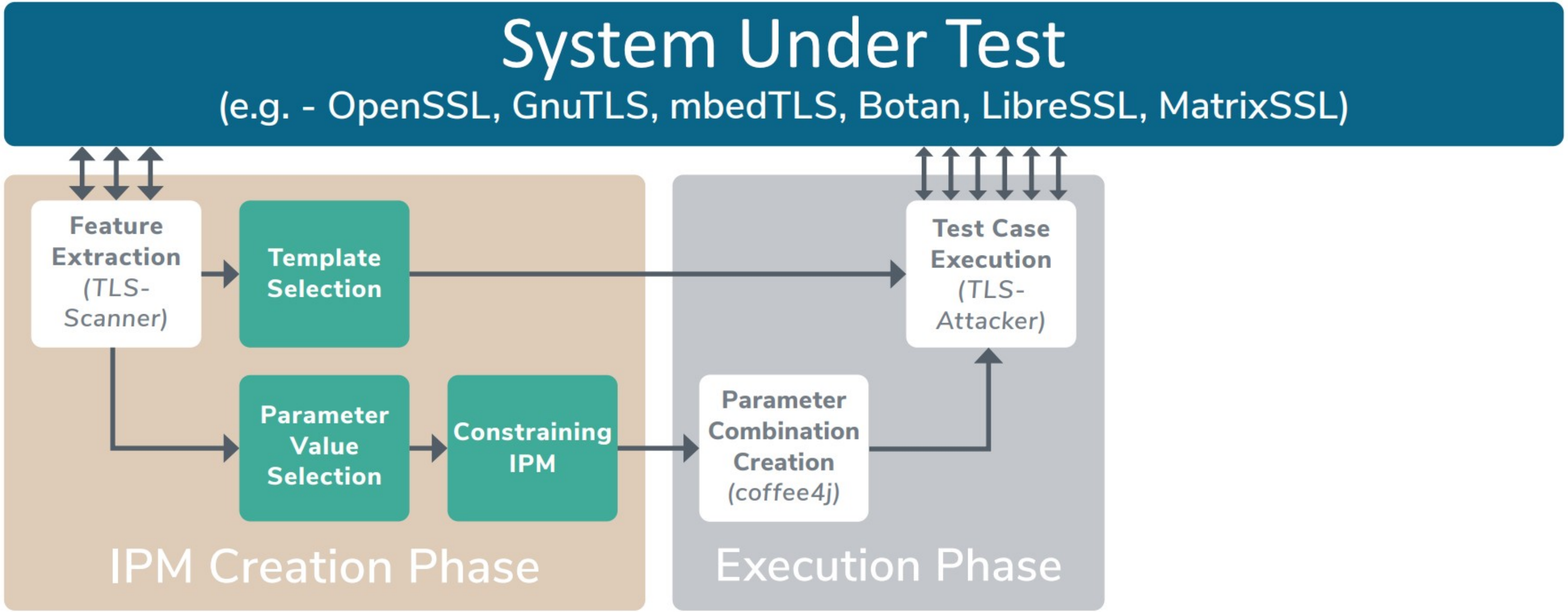
Execution


System Under Test
(e.g. - OpenSSL, GnuTLS, mbedTLS, Botan, LibreSSL, MatrixSSL)
Feature Extraction (TLS-Scanner)
Template Selection
Parameter Value Selection
Constraining IPM
IPM Creation Phase

Execution


System Under Test
(e.g. - OpenSSL, GnuTLS, mbedTLS, Botan, LibreSSL, MatrixSSL)
Feature Extraction (TLS-Scanner)
Template Selection
Parameter Value Selection
Constraining IPM
Parameter Combination Creation (coffee4j)
IPM Creation Phase
Execution Phase

Execution



System Under Test

(e.g. - OpenSSL, GnuTLS, mbedTLS, Botan, LibreSSL, MatrixSSL)

Feature Extraction (TLS-Scanner)

Template Selection

Parameter Value Selection

Constraining IPM

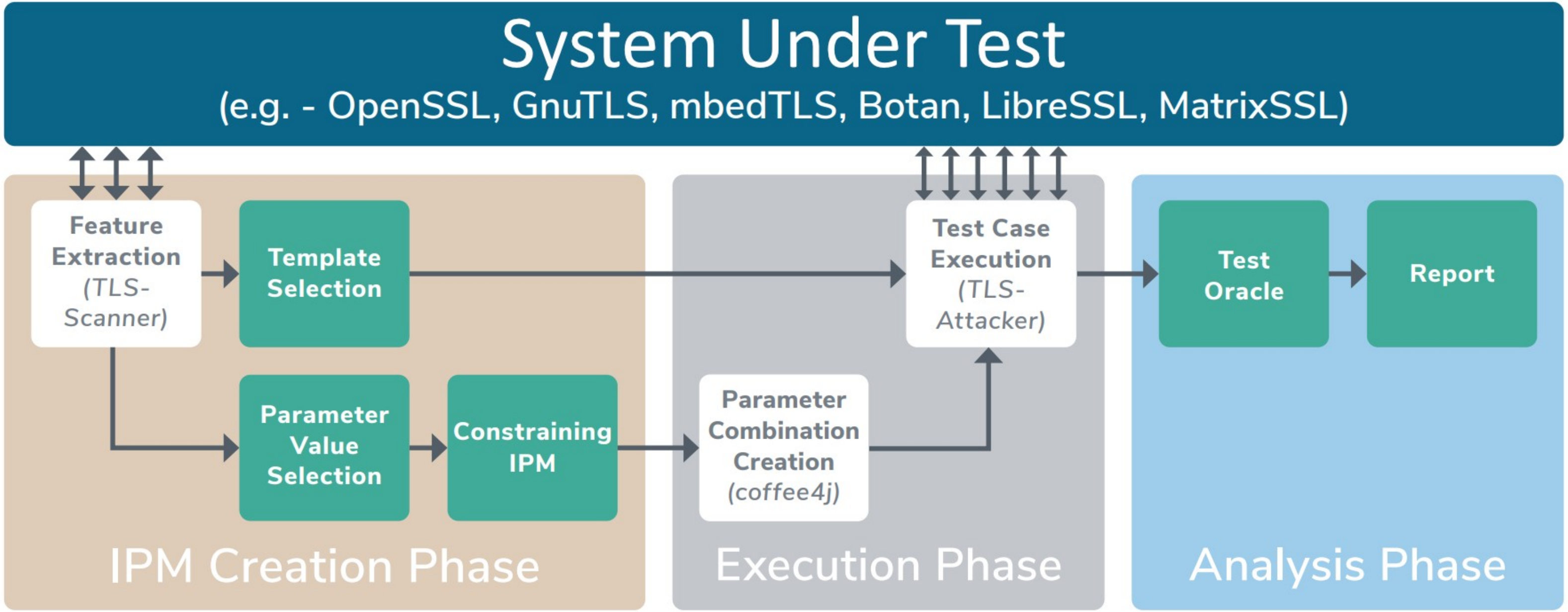
Parameter Combination Creation (coffee4j)

Test Case Execution (TLS-Attacker)

IPM Creation Phase

Execution Phase

Execution



System Under Test

(e.g. - OpenSSL, GnuTLS, mbedTLS, Botan, LibreSSL, MatrixSSL)

Feature Extraction (TLS-Scanner)

Template Selection

Parameter Value Selection

Constraining IPM

IPM Creation Phase

Parameter Combination Creation (coffee4j)

Test Case Execution (TLS-Attacker)

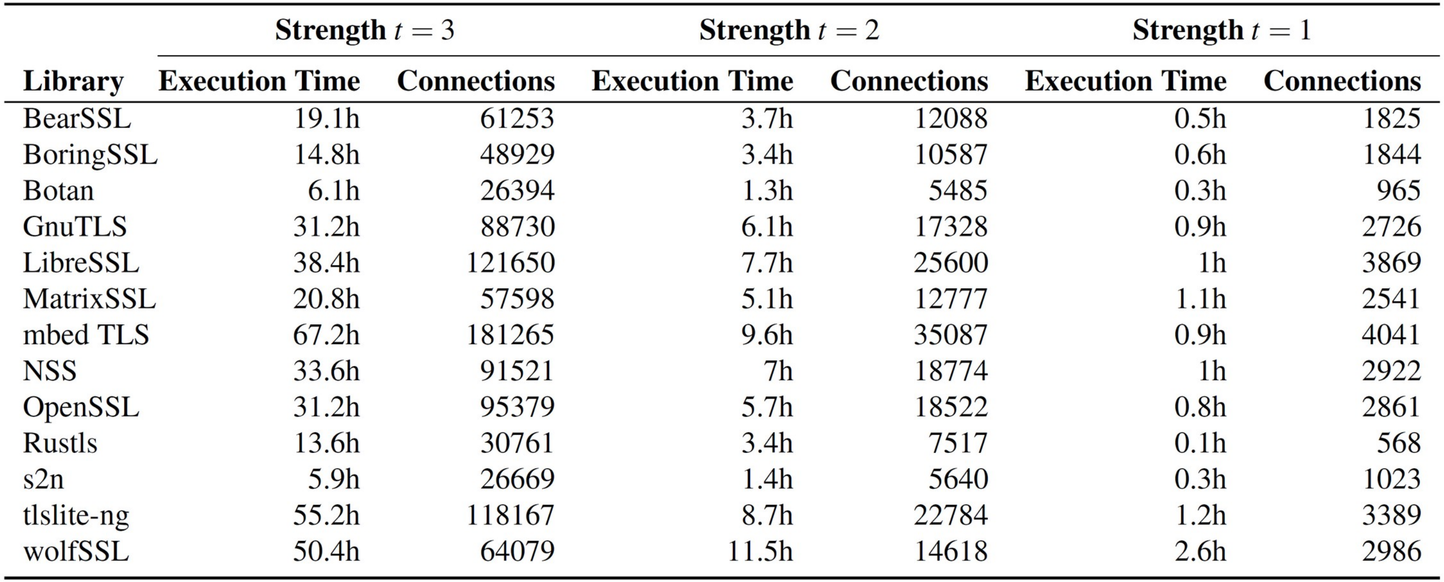
Execution Phase

Test Oracle

Report

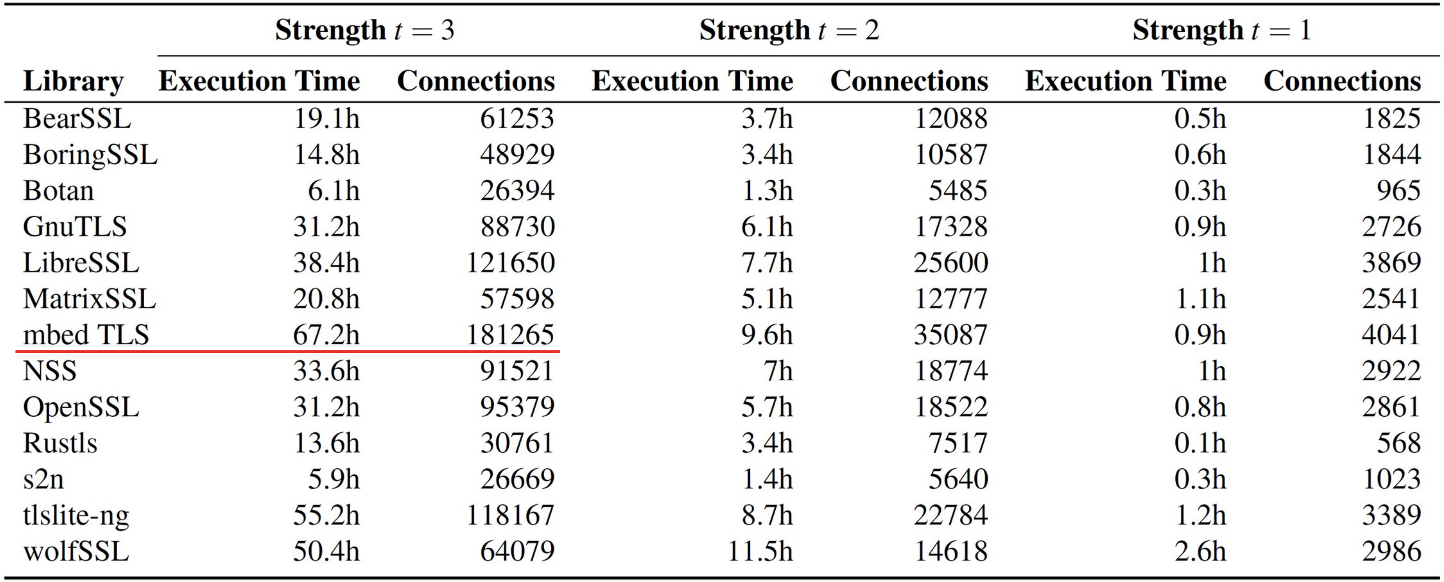
Analysis Phase

Performance Evaluation



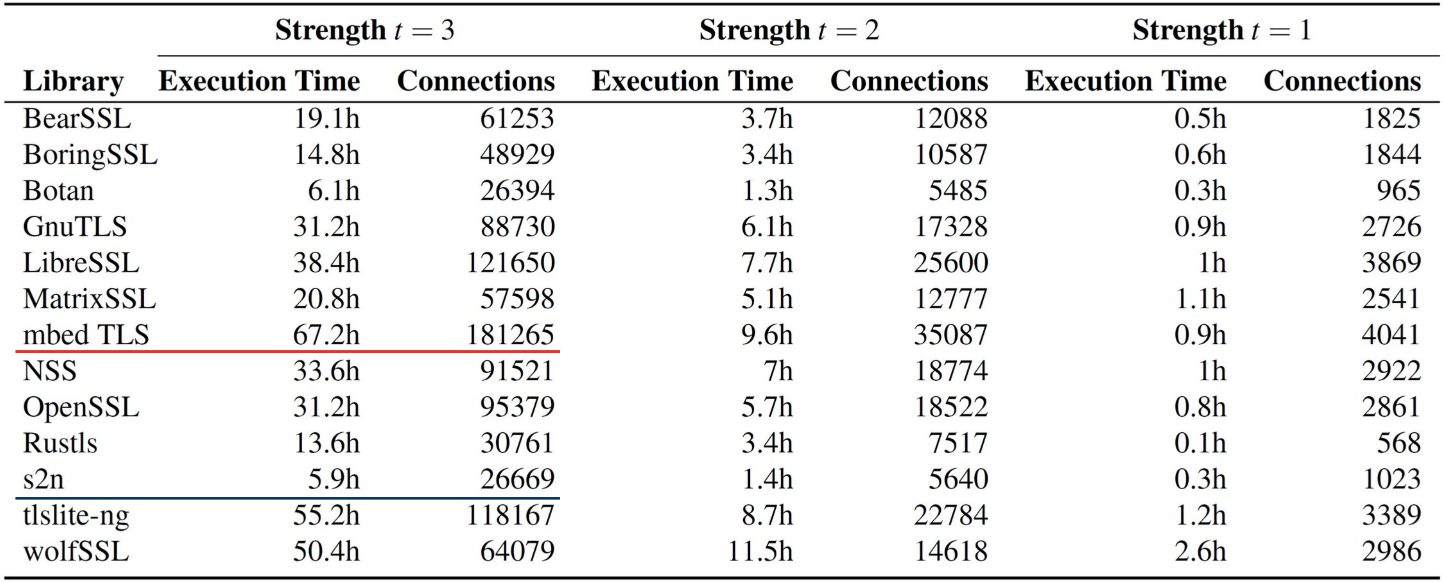
	Strength $t = 3$		Strength $t = 2$		Strength $t = 1$	
Library	**Execution Time**	**Connections**	**Execution Time**	**Connections**	**Execution Time**	**Connections**
BearSSL	19.1h	61253	3.7h	12088	0.5h	1825
BoringSSL	14.8h	48929	3.4h	10587	0.6h	1844
Botan	6.1h	26394	1.3h	5485	0.3h	965
GnuTLS	31.2h	88730	6.1h	17328	0.9h	2726
LibreSSL	38.4h	121650	7.7h	25600	1h	3869
MatrixSSL	20.8h	57598	5.1h	12777	1.1h	2541
mbed TLS	67.2h	181265	9.6h	35087	0.9h	4041
NSS	33.6h	91521	7h	18774	1h	2922
OpenSSL	31.2h	95379	5.7h	18522	0.8h	2861
Rustls	13.6h	30761	3.4h	7517	0.1h	568
s2n	5.9h	26669	1.4h	5640	0.3h	1023
tlslite-ng	55.2h	118167	8.7h	22784	1.2h	3389
wolfSSL	50.4h	64079	11.5h	14618	2.6h	2986

Performance Evaluation



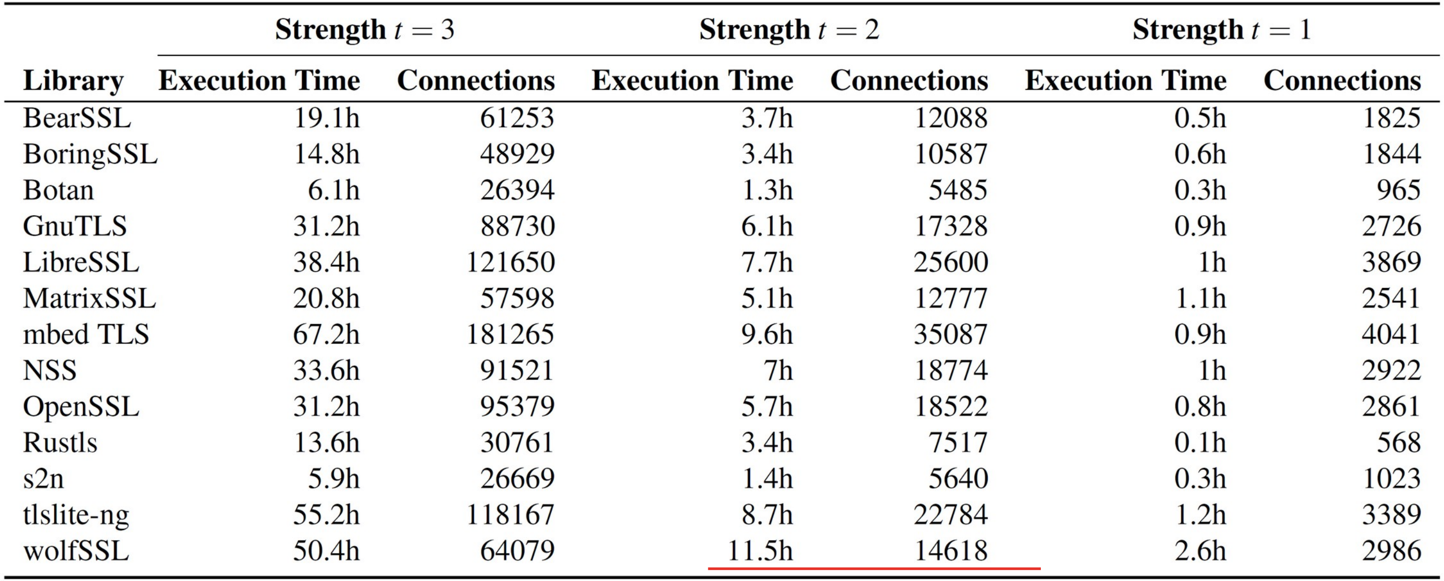
	Strength $t = 3$		Strength $t = 2$		Strength $t = 1$	
Library	**Execution Time**	**Connections**	**Execution Time**	**Connections**	**Execution Time**	**Connections**
BearSSL	19.1h	61253	3.7h	12088	0.5h	1825
BoringSSL	14.8h	48929	3.4h	10587	0.6h	1844
Botan	6.1h	26394	1.3h	5485	0.3h	965
GnuTLS	31.2h	88730	6.1h	17328	0.9h	2726
LibreSSL	38.4h	121650	7.7h	25600	1h	3869
MatrixSSL	20.8h	57598	5.1h	12777	1.1h	2541
mbed TLS	67.2h	181265	9.6h	35087	0.9h	4041
NSS	33.6h	91521	7h	18774	1h	2922
OpenSSL	31.2h	95379	5.7h	18522	0.8h	2861
Rustls	13.6h	30761	3.4h	7517	0.1h	568
s2n	5.9h	26669	1.4h	5640	0.3h	1023
tlslite-ng	55.2h	118167	8.7h	22784	1.2h	3389
wolfSSL	50.4h	64079	11.5h	14618	2.6h	2986

Performance Evaluation



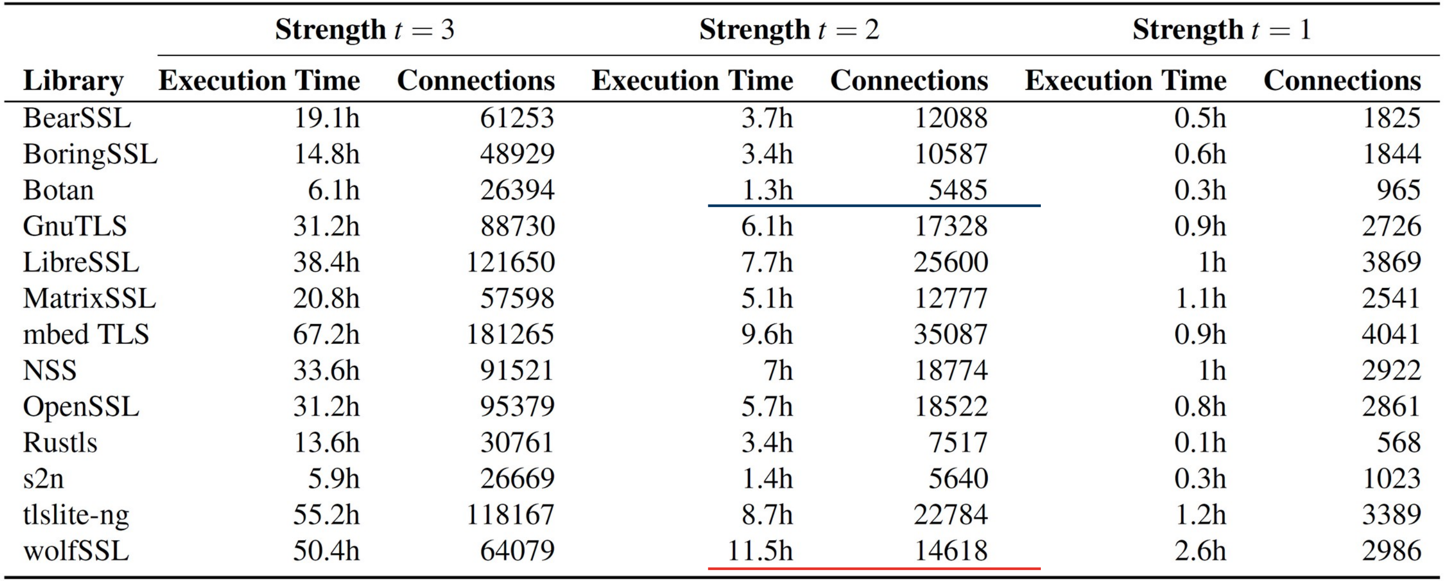
	Strength $t = 3$		Strength $t = 2$		Strength $t = 1$	
Library	**Execution Time**	**Connections**	**Execution Time**	**Connections**	**Execution Time**	**Connections**
BearSSL	19.1h	61253	3.7h	12088	0.5h	1825
BoringSSL	14.8h	48929	3.4h	10587	0.6h	1844
Botan	6.1h	26394	1.3h	5485	0.3h	965
GnuTLS	31.2h	88730	6.1h	17328	0.9h	2726
LibreSSL	38.4h	121650	7.7h	25600	1h	3869
MatrixSSL	20.8h	57598	5.1h	12777	1.1h	2541
mbed TLS	67.2h	181265	9.6h	35087	0.9h	4041
NSS	33.6h	91521	7h	18774	1h	2922
OpenSSL	31.2h	95379	5.7h	18522	0.8h	2861
Rustls	13.6h	30761	3.4h	7517	0.1h	568
s2n	5.9h	26669	1.4h	5640	0.3h	1023
tlslite-ng	55.2h	118167	8.7h	22784	1.2h	3389
wolfSSL	50.4h	64079	11.5h	14618	2.6h	2986

Performance Evaluation



	Strength $t = 3$		Strength $t = 2$		Strength $t = 1$	
Library	**Execution Time**	**Connections**	**Execution Time**	**Connections**	**Execution Time**	**Connections**
BearSSL	19.1h	61253	3.7h	12088	0.5h	1825
BoringSSL	14.8h	48929	3.4h	10587	0.6h	1844
Botan	6.1h	26394	1.3h	5485	0.3h	965
GnuTLS	31.2h	88730	6.1h	17328	0.9h	2726
LibreSSL	38.4h	121650	7.7h	25600	1h	3869
MatrixSSL	20.8h	57598	5.1h	12777	1.1h	2541
mbed TLS	67.2h	181265	9.6h	35087	0.9h	4041
NSS	33.6h	91521	7h	18774	1h	2922
OpenSSL	31.2h	95379	5.7h	18522	0.8h	2861
Rustls	13.6h	30761	3.4h	7517	0.1h	568
s2n	5.9h	26669	1.4h	5640	0.3h	1023
tlslite-ng	55.2h	118167	8.7h	22784	1.2h	3389
wolfSSL	50.4h	64079	11.5h	14618	2.6h	2986

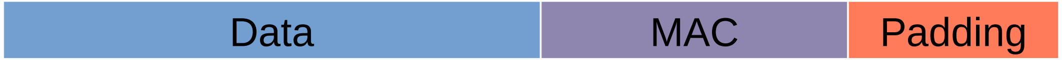
Performance Evaluation



	Strength $t = 3$		Strength $t = 2$		Strength $t = 1$	
Library	**Execution Time**	**Connections**	**Execution Time**	**Connections**	**Execution Time**	**Connections**
BearSSL	19.1h	61253	3.7h	12088	0.5h	1825
BoringSSL	14.8h	48929	3.4h	10587	0.6h	1844
Botan	6.1h	26394	1.3h	5485	0.3h	965
GnuTLS	31.2h	88730	6.1h	17328	0.9h	2726
LibreSSL	38.4h	121650	7.7h	25600	1h	3869
MatrixSSL	20.8h	57598	5.1h	12777	1.1h	2541
mbed TLS	67.2h	181265	9.6h	35087	0.9h	4041
NSS	33.6h	91521	7h	18774	1h	2922
OpenSSL	31.2h	95379	5.7h	18522	0.8h	2861
Rustls	13.6h	30761	3.4h	7517	0.1h	568
s2n	5.9h	26669	1.4h	5640	0.3h	1023
tlslite-ng	55.2h	118167	8.7h	22784	1.2h	3389
wolfSSL	50.4h	64079	11.5h	14618	2.6h	2986

Padding Oracles in TLS

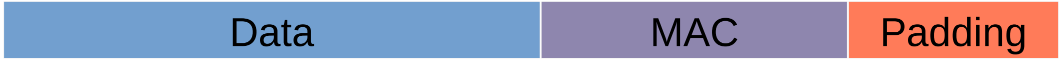
- TLS uses MAC-then-Encrypt



Data MAC Padding

Padding Oracles in TLS

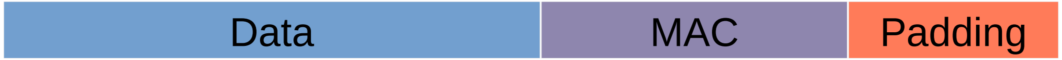
- TLS uses MAC-then-Encrypt



- **Invalid padding must be indistinguishable from invalid MAC**

Padding Oracles in TLS

- TLS uses MAC-then-Encrypt


Data
MAC
Padding

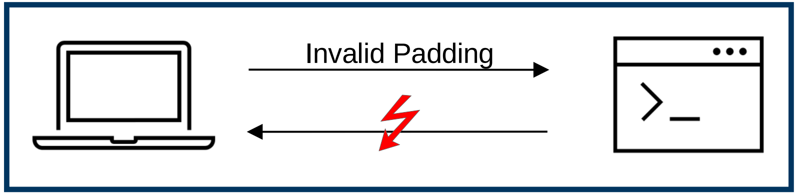
- **Invalid padding must be indistinguishable from invalid MAC**

- If padding is invalid, implementation must proceed to compute MAC before failing

- Subsequently, send *Bad Record MAC Alert*

Padding Oracle in MatrixSSL Client

- Invalid padding led to closed TCP connection for SHA-256 HMAC

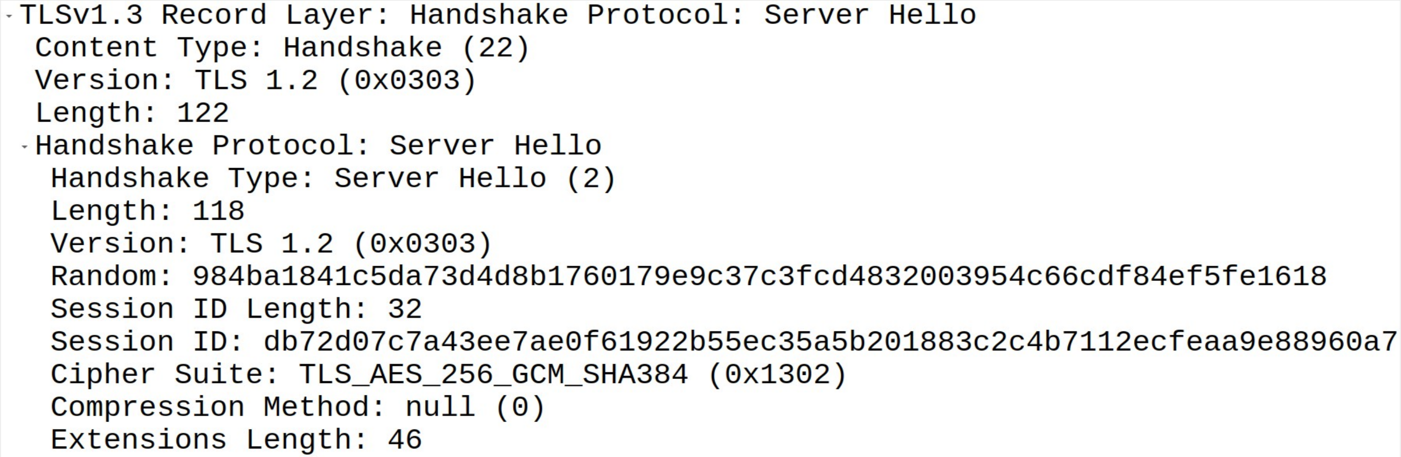


- Caused by segmentation fault due to uninitialized HMAC

- Distinguishable from MAC failures and thus exploitable

DoS in MatrixSSL Client

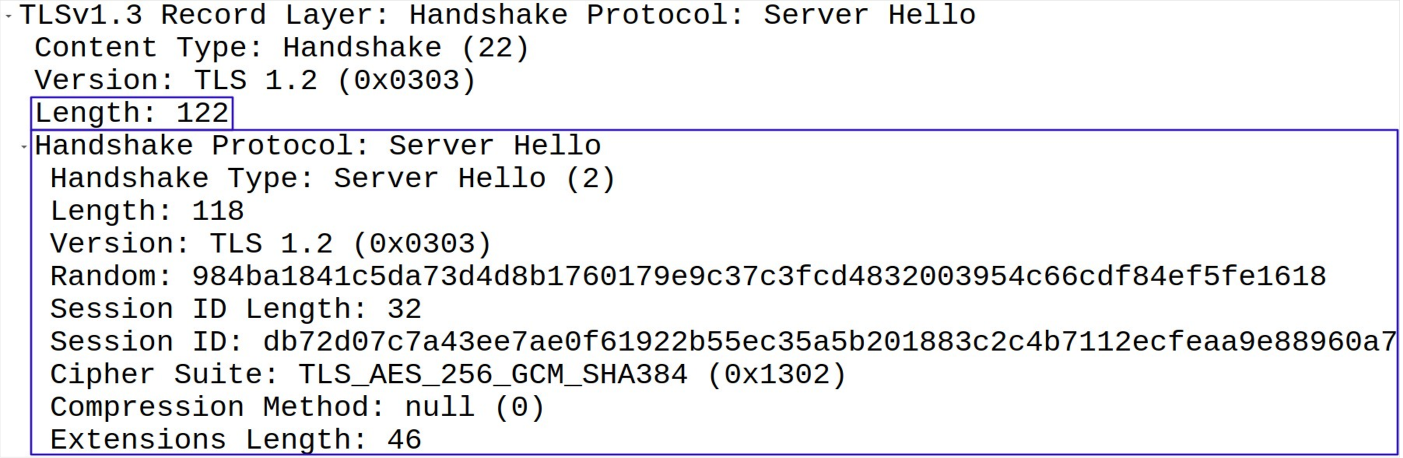
- Send message with contradicting length fields



```
TLSv1.3 Record Layer: Handshake Protocol: Server Hello
 Content Type: Handshake (22)
 Version: TLS 1.2 (0x0303)
 Length: 122
 Handshake Protocol: Server Hello
  Handshake Type: Server Hello (2)
  Length: 118
  Version: TLS 1.2 (0x0303)
  Random: 984ba1841c5da73d4d8b1760179e9c37c3fcd4832003954c66cdf84ef5fe1618
  Session ID Length: 32
  Session ID: db72d07c7a43ee7ae0f61922b55ec35a5b201883c2c4b7112ecfeaa9e88960a7
  Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
  Compression Method: null (0)
  Extensions Length: 46
```

DoS in MatrixSSL Client

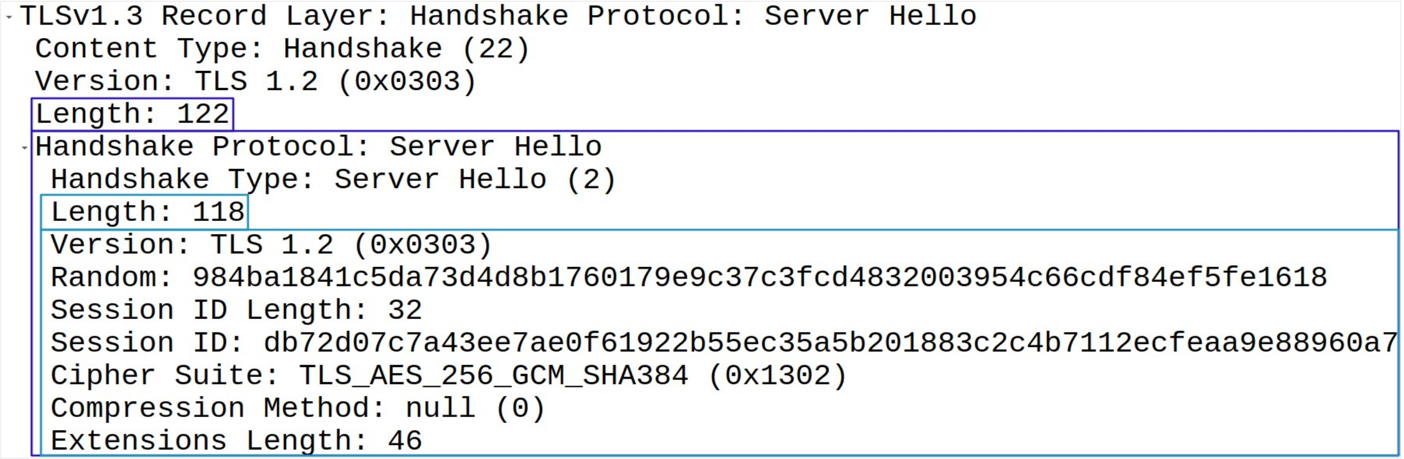
- Send message with contradicting length fields



```
- TLSv1.3 Record Layer: Handshake Protocol: Server Hello
   Content Type: Handshake (22)
   Version: TLS 1.2 (0x0303)
   Length: 122
 - Handshake Protocol: Server Hello
     Handshake Type: Server Hello (2)
     Length: 118
     Version: TLS 1.2 (0x0303)
     Random: 984ba1841c5da73d4d8b1760179e9c37c3fcd4832003954c66cdf84ef5fe1618
     Session ID Length: 32
     Session ID: db72d07c7a43ee7ae0f61922b55ec35a5b201883c2c4b7112ecfeaa9e88960a7
     Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
     Compression Method: null (0)
     Extensions Length: 46
```

DoS in MatrixSSL Client

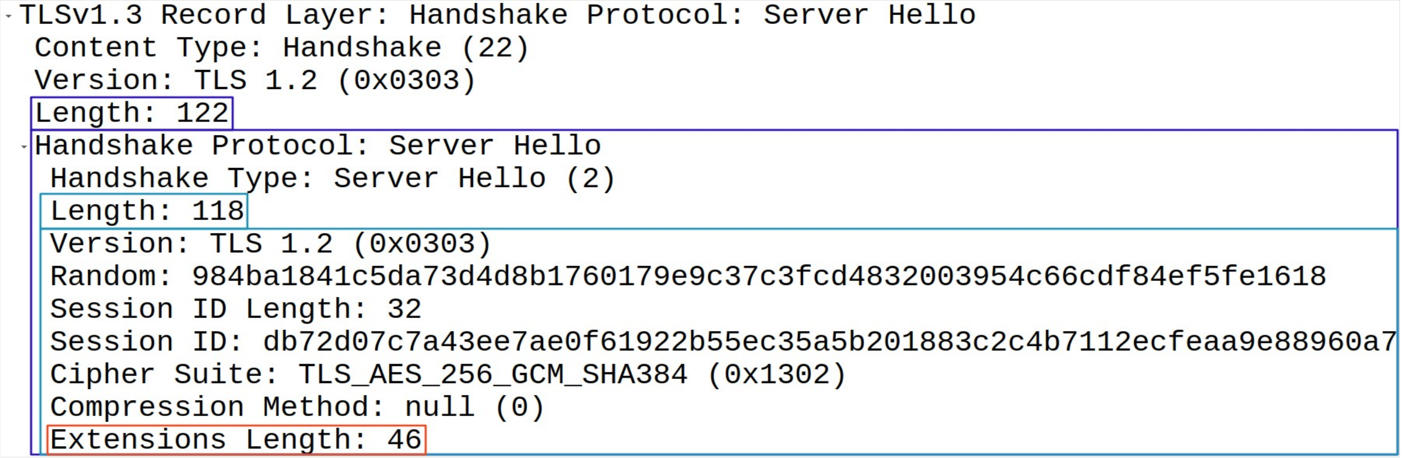
- Send message with contradicting length fields



```
TLSv1.3 Record Layer: Handshake Protocol: Server Hello
 Content Type: Handshake (22)
 Version: TLS 1.2 (0x0303)
 Length: 122
 Handshake Protocol: Server Hello
  Handshake Type: Server Hello (2)
  Length: 118
  Version: TLS 1.2 (0x0303)
  Random: 984ba1841c5da73d4d8b1760179e9c37c3fcd4832003954c66cdf84ef5fe1618
  Session ID Length: 32
  Session ID: db72d07c7a43ee7ae0f61922b55ec35a5b201883c2c4b7112ecfeaa9e88960a7
  Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
  Compression Method: null (0)
  Extensions Length: 46
```

DoS in MatrixSSL Client

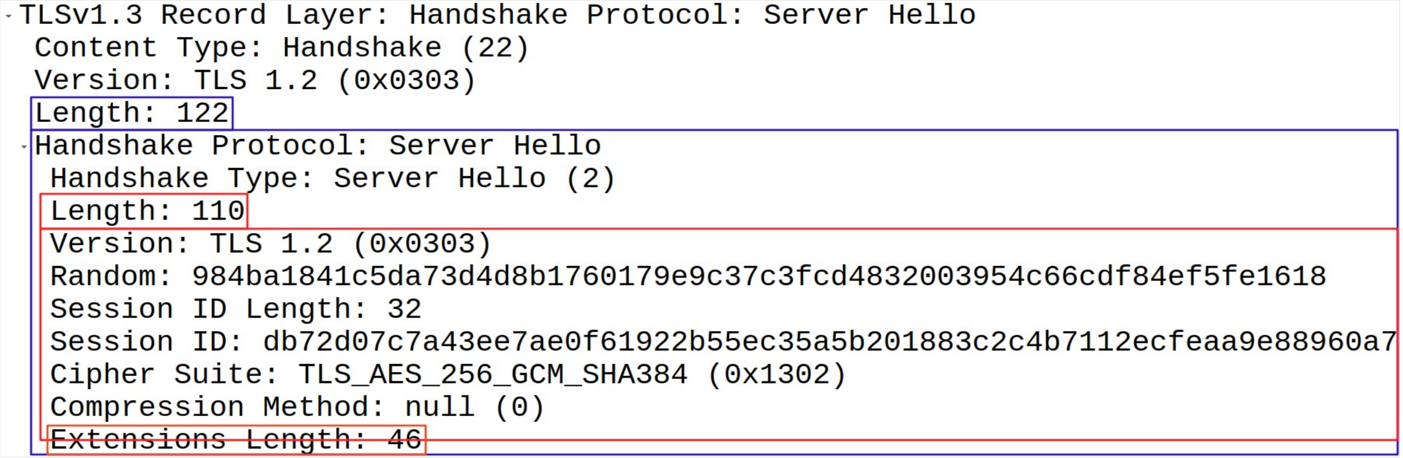
- Send message with contradicting length fields



```
▾TLSv1.3 Record Layer: Handshake Protocol: Server Hello
  Content Type: Handshake (22)
  Version: TLS 1.2 (0x0303)
  Length: 122
 ▾Handshake Protocol: Server Hello
   Handshake Type: Server Hello (2)
   Length: 118
   Version: TLS 1.2 (0x0303)
   Random: 984ba1841c5da73d4d8b1760179e9c37c3fcd4832003954c66cdf84ef5fe1618
   Session ID Length: 32
   Session ID: db72d07c7a43ee7ae0f61922b55ec35a5b201883c2c4b7112ecfeaa9e88960a7
   Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
   Compression Method: null (0)
   Extensions Length: 46
```

DoS in MatrixSSL Client

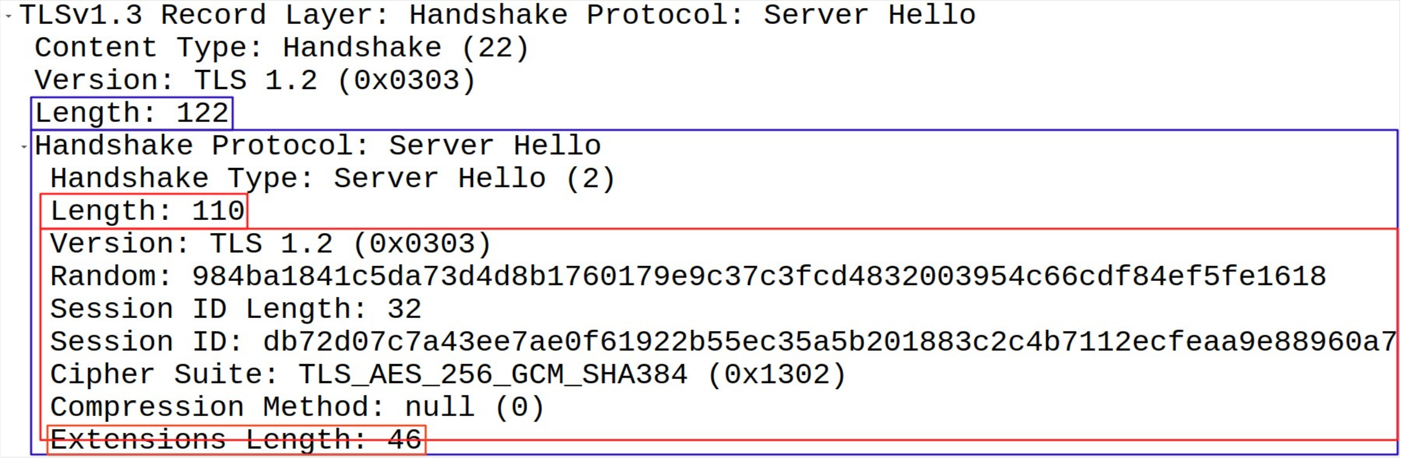
- Send message with contradicting length fields



```
TLSv1.3 Record Layer: Handshake Protocol: Server Hello
 Content Type: Handshake (22)
 Version: TLS 1.2 (0x0303)
 Length: 122
 Handshake Protocol: Server Hello
  Handshake Type: Server Hello (2)
  Length: 110
  Version: TLS 1.2 (0x0303)
  Random: 984ba1841c5da73d4d8b1760179e9c37c3fcd4832003954c66cdf84ef5fe1618
  Session ID Length: 32
  Session ID: db72d07c7a43ee7ae0f61922b55ec35a5b201883c2c4b7112ecfeaa9e88960a7
  Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
  Compression Method: null (0)
  Extensions Length: 46
```

DoS in MatrixSSL Client

- Send message with contradicting length fields

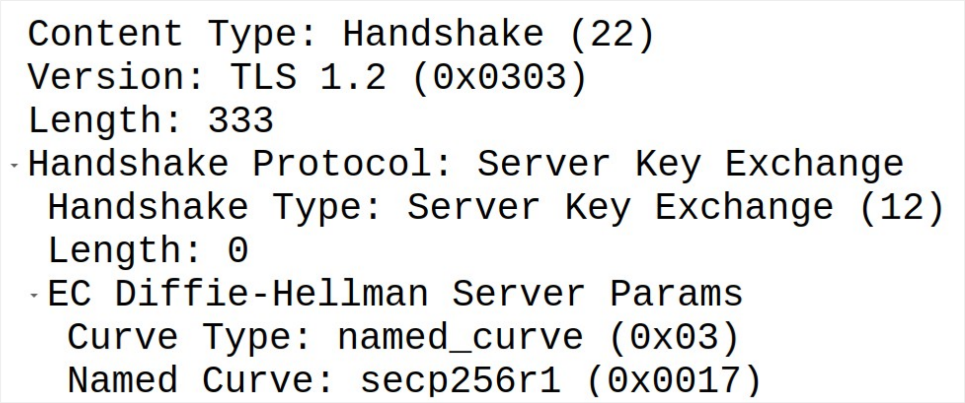


```
TLSv1.3 Record Layer: Handshake Protocol: Server Hello
 Content Type: Handshake (22)
 Version: TLS 1.2 (0x0303)
 Length: 122
 Handshake Protocol: Server Hello
  Handshake Type: Server Hello (2)
  Length: 110
  Version: TLS 1.2 (0x0303)
  Random: 984ba1841c5da73d4d8b1760179e9c37c3fcd4832003954c66cdf84ef5fe1618
  Session ID Length: 32
  Session ID: db72d07c7a43ee7ae0f61922b55ec35a5b201883c2c4b7112ecfeaa9e88960a7
  Cipher Suite: TLS_AES_256_GCM_SHA384 (0x1302)
  Compression Method: null (0)
  Extensions Length: 46
```

- Causes CPU usage to rise and MatrixSSL becomes unresponsive in TLS 1.3

MatrixSSL Lengthfield Bug in TLS 1.2

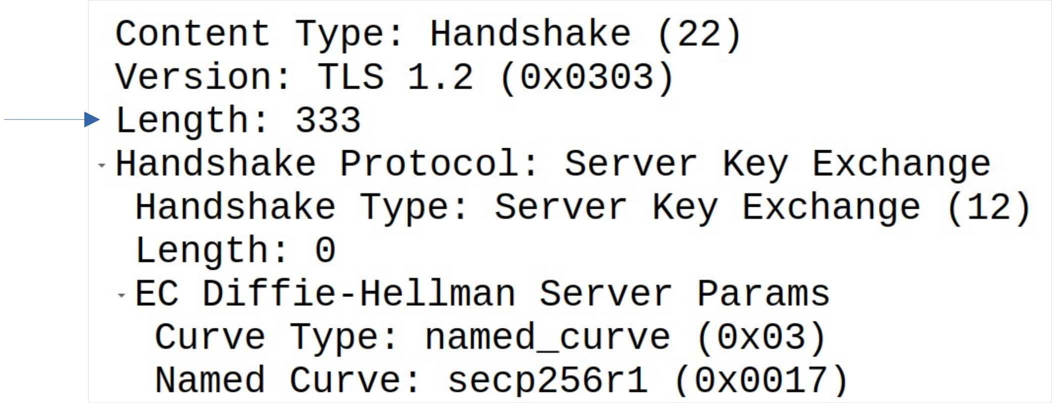
- Send message with content but set content length to 0



```
Content Type: Handshake (22)
Version: TLS 1.2 (0x0303)
Length: 333
Handshake Protocol: Server Key Exchange
 Handshake Type: Server Key Exchange (12)
 Length: 0
 EC Diffie-Hellman Server Params
  Curve Type: named_curve (0x03)
  Named Curve: secp256r1 (0x0017)
```

MatrixSSL Lengthfield Bug in TLS 1.2

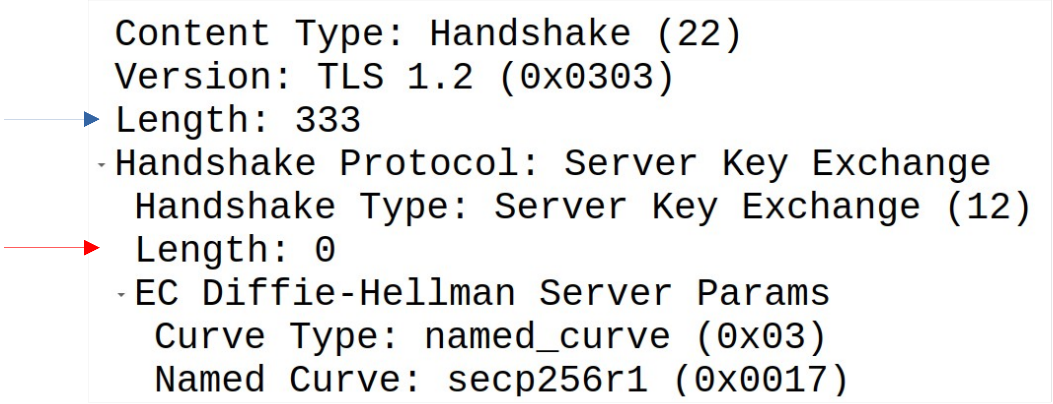
- Send message with content but set content length to 0



```
Content Type: Handshake (22)
Version: TLS 1.2 (0x0303)
Length: 333
Handshake Protocol: Server Key Exchange
 Handshake Type: Server Key Exchange (12)
 Length: 0
 EC Diffie-Hellman Server Params
  Curve Type: named_curve (0x03)
  Named Curve: secp256r1 (0x0017)
```

MatrixSSL Lengthfield Bug in TLS 1.2

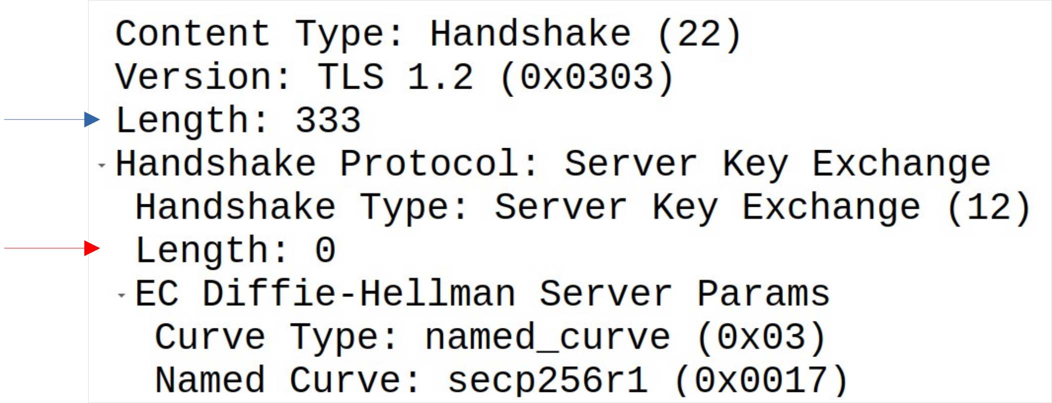
- Send message with content but set content length to 0



```
Content Type: Handshake (22)
Version: TLS 1.2 (0x0303)
Length: 333
Handshake Protocol: Server Key Exchange
 Handshake Type: Server Key Exchange (12)
 Length: 0
 EC Diffie-Hellman Server Params
  Curve Type: named_curve (0x03)
  Named Curve: secp256r1 (0x0017)
```

MatrixSSL Lengthfield Bug in TLS 1.2

- Send message with content but set content length to 0

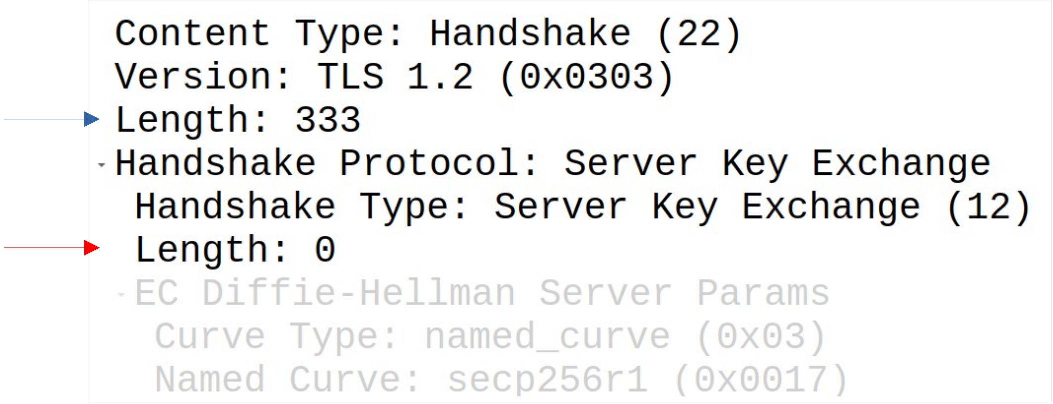


```
Content Type: Handshake (22)
Version: TLS 1.2 (0x0303)
Length: 333
Handshake Protocol: Server Key Exchange
 Handshake Type: Server Key Exchange (12)
 Length: 0
 EC Diffie-Hellman Server Params
  Curve Type: named_curve (0x03)
  Named Curve: secp256r1 (0x0017)
```

- Message will be parsed correctly but content won‘t affect session transcript in *Finished* message

MatrixSSL Lengthfield Bug in TLS 1.2

- Send message with content but set content length to 0

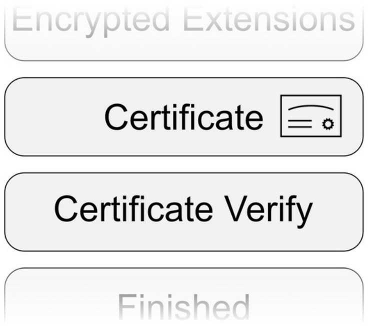


```
Content Type: Handshake (22)
Version: TLS 1.2 (0x0303)
Length: 333
Handshake Protocol: Server Key Exchange
 Handshake Type: Server Key Exchange (12)
 Length: 0
 EC Diffie-Hellman Server Params
  Curve Type: named_curve (0x03)
  Named Curve: secp256r1 (0x0017)
```

- Message will be parsed correctly but content won‘t affect session transcript in *Finished* message

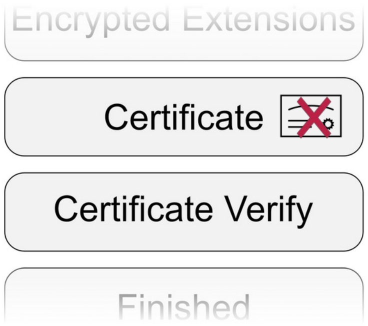
Authentication bypass for wolfSSL in TLS 1.3

- Usually, *Certificate* is used to verify signature in *Certificate Verify*



Authentication bypass for wolfSSL in TLS 1.3

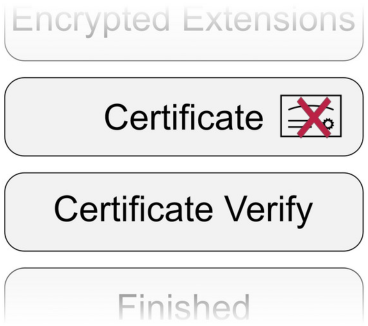
- Usually, *Certificate* is used to verify signature in *Certificate Verify*



- If *Certificate* message is emtpy, wolfSSL accepted any *Certificate Verify*

Authentication bypass for wolfSSL in TLS 1.3

- Usually, *Certificate* is used to verify signature in *Certificate Verify*

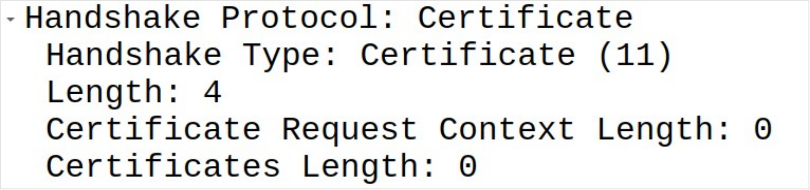


Encrypted Extensions

Certificate

Certificate Verify

Finished

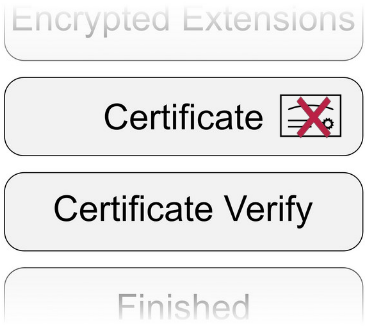


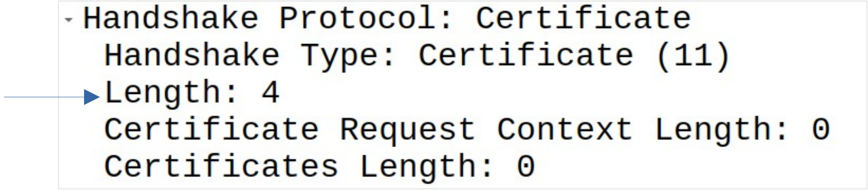
```
- Handshake Protocol: Certificate
    Handshake Type: Certificate (11)
    Length: 4
    Certificate Request Context Length: 0
    Certificates Length: 0
```

- If *Certificate* message is emtpy, wolfSSL accepted any *Certificate Verify*

Authentication bypass for wolfSSL in TLS 1.3

- Usually, *Certificate* is used to verify signature in *Certificate Verify*



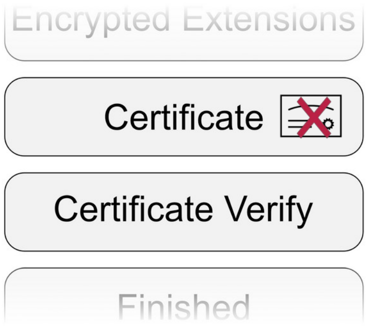


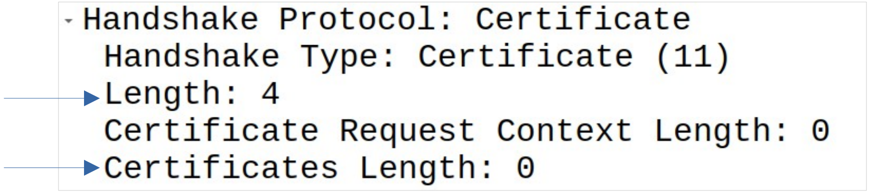
```
▾ Handshake Protocol: Certificate
    Handshake Type: Certificate (11)
    Length: 4
    Certificate Request Context Length: 0
    Certificates Length: 0
```

- If *Certificate* message is emtpy, wolfSSL accepted any *Certificate Verify*

Authentication bypass for wolfSSL in TLS 1.3

- Usually, *Certificate* is used to verify signature in *Certificate Verify*



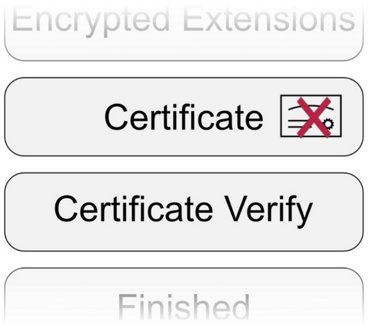


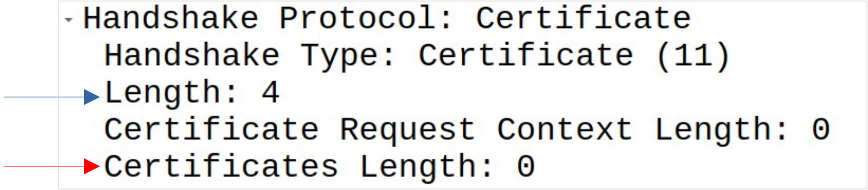
```
▾ Handshake Protocol: Certificate
    Handshake Type: Certificate (11)
    Length: 4
    Certificate Request Context Length: 0
    Certificates Length: 0
```

- If *Certificate* message is emtpy, wolfSSL accepted any *Certificate Verify*

Authentication bypass for wolfSSL in TLS 1.3

- Usually, *Certificate* is used to verify signature in *Certificate Verify*

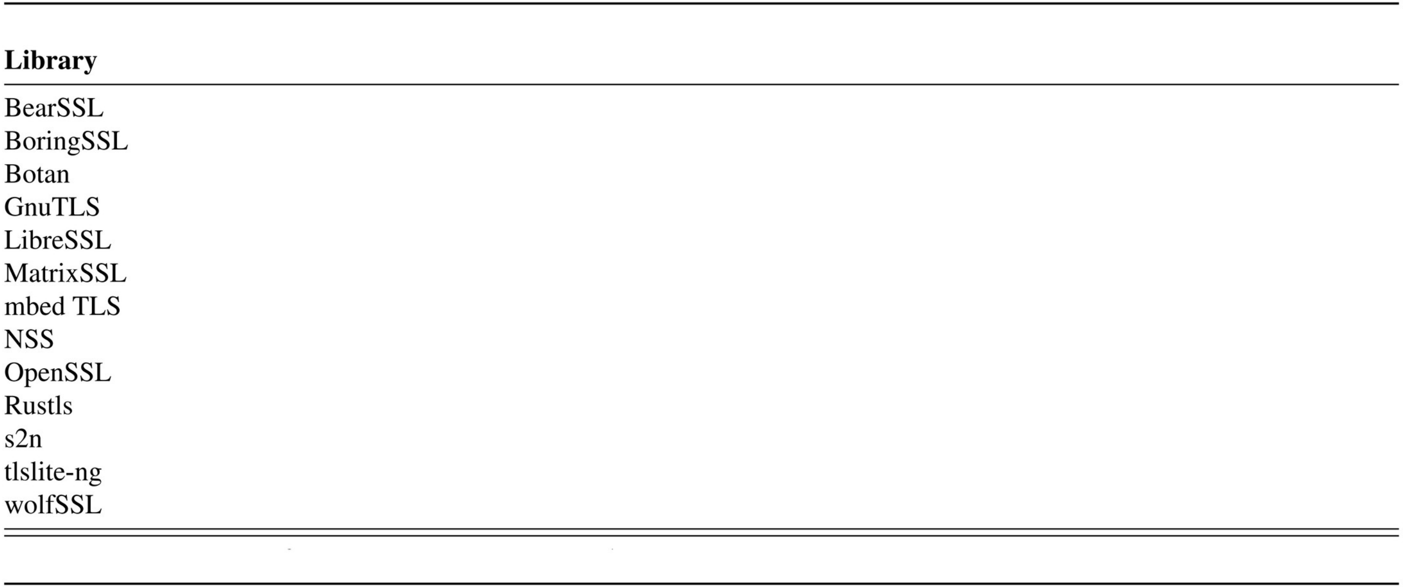




```
Handshake Protocol: Certificate
  Handshake Type: Certificate (11)
  Length: 4
  Certificate Request Context Length: 0
  Certificates Length: 0
```

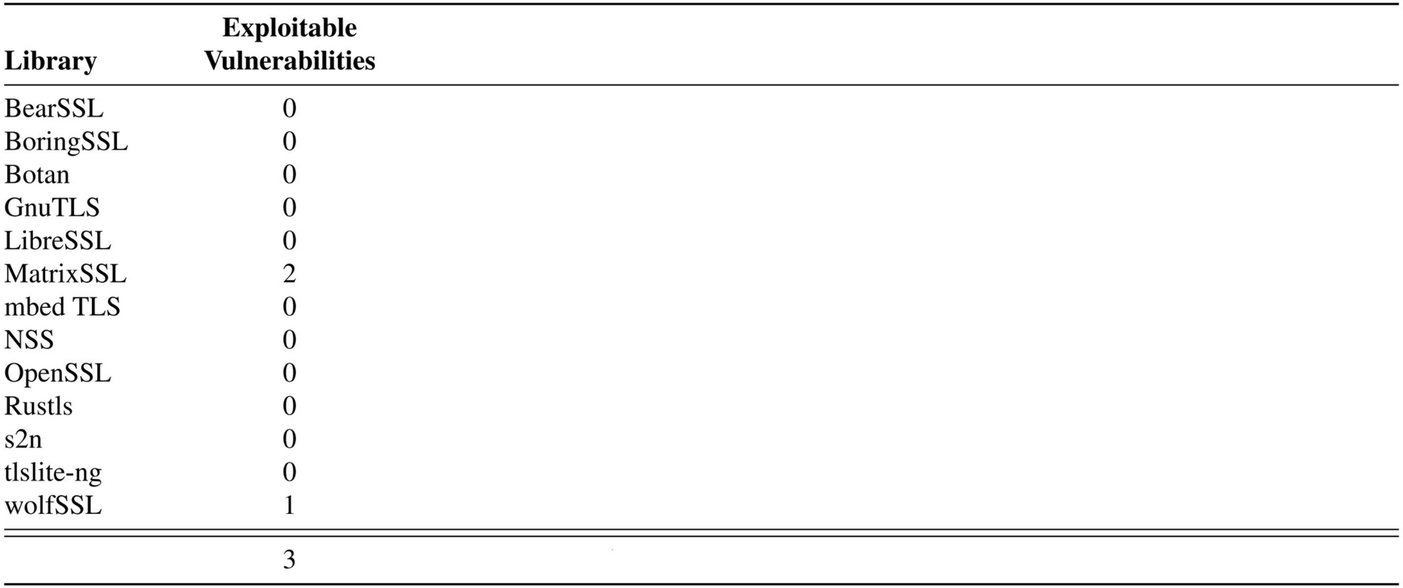
- If *Certificate* message is emtpy, wolfSSL accepted any *Certificate Verify*

239 RFC violations



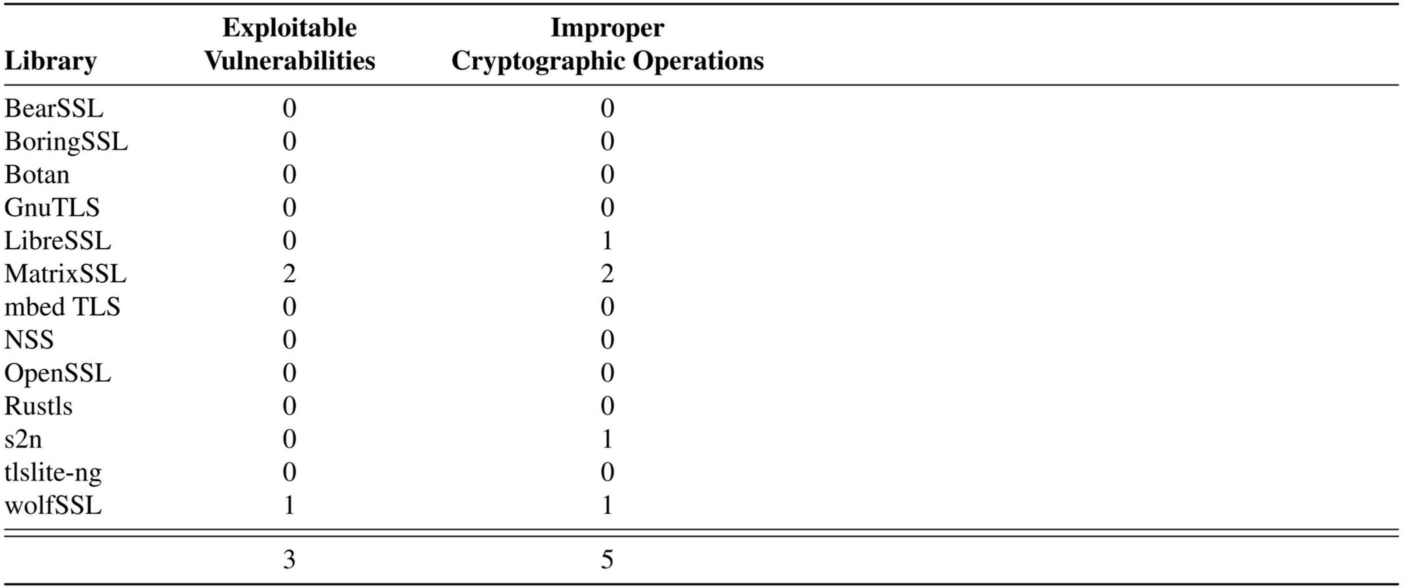
Library
BearSSL
BoringSSL
Botan
GnuTLS
LibreSSL
MatrixSSL
mbed TLS
NSS
OpenSSL
Rustls
s2n
tlslite-ng
wolfSSL

239 RFC violations



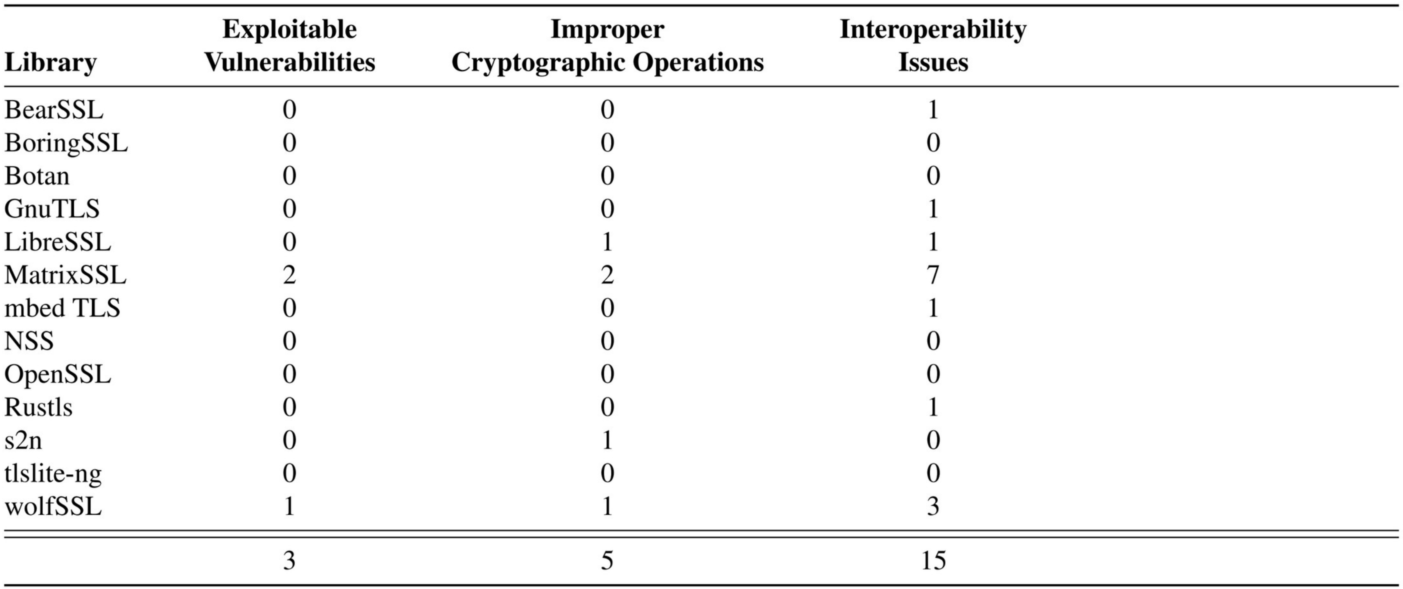
Library	Exploitable Vulnerabilities
BearSSL	0
BoringSSL	0
Botan	0
GnuTLS	0
LibreSSL	0
MatrixSSL	2
mbed TLS	0
NSS	0
OpenSSL	0
Rustls	0
s2n	0
tlslite-ng	0
wolfSSL	1
	3

239 RFC violations



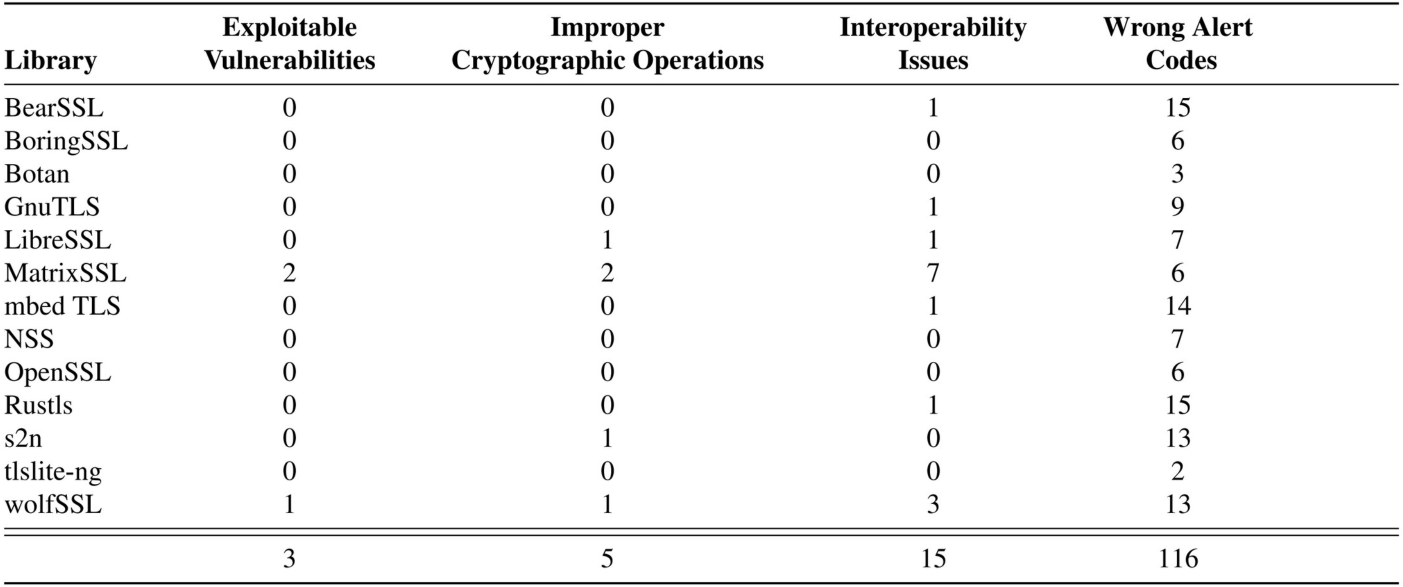
Library	Exploitable Vulnerabilities	Improper Cryptographic Operations
BearSSL	0	0
BoringSSL	0	0
Botan	0	0
GnuTLS	0	0
LibreSSL	0	1
MatrixSSL	2	2
mbed TLS	0	0
NSS	0	0
OpenSSL	0	0
Rustls	0	0
s2n	0	1
tlslite-ng	0	0
wolfSSL	1	1
	3	5

239 RFC violations



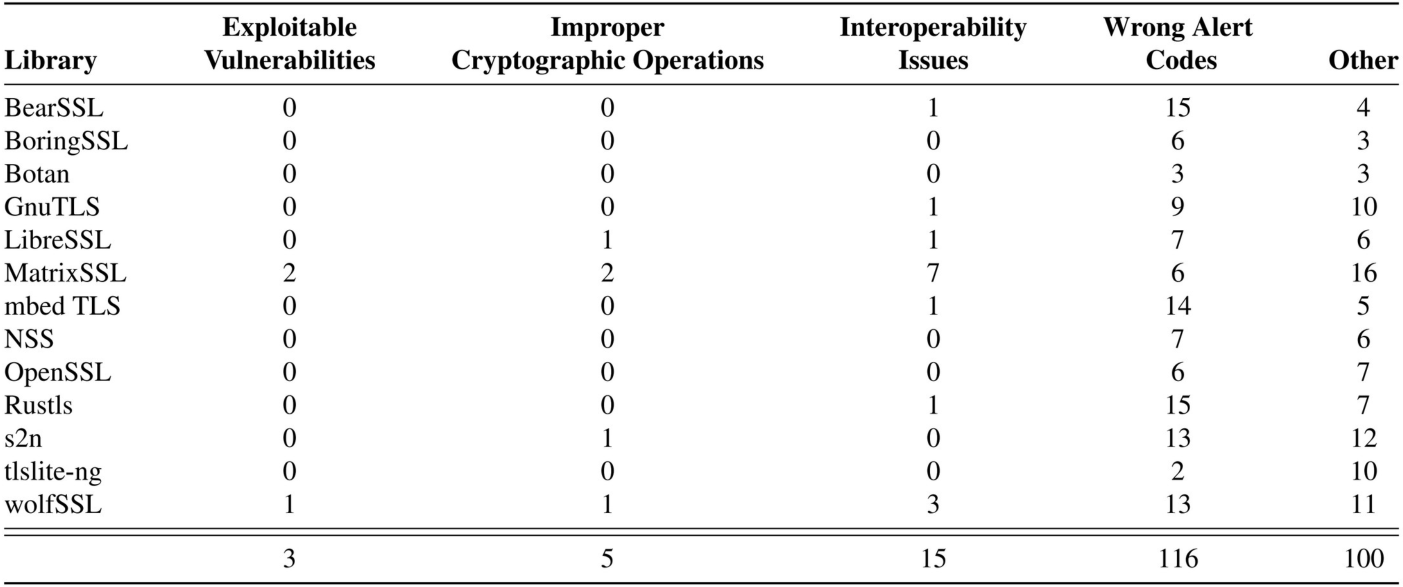
Library	Exploitable Vulnerabilities	Improper Cryptographic Operations	Interoperability Issues
BearSSL	0	0	1
BoringSSL	0	0	0
Botan	0	0	0
GnuTLS	0	0	1
LibreSSL	0	1	1
MatrixSSL	2	2	7
mbed TLS	0	0	1
NSS	0	0	0
OpenSSL	0	0	0
Rustls	0	0	1
s2n	0	1	0
tlslite-ng	0	0	0
wolfSSL	1	1	3
	3	5	15

239 RFC violations



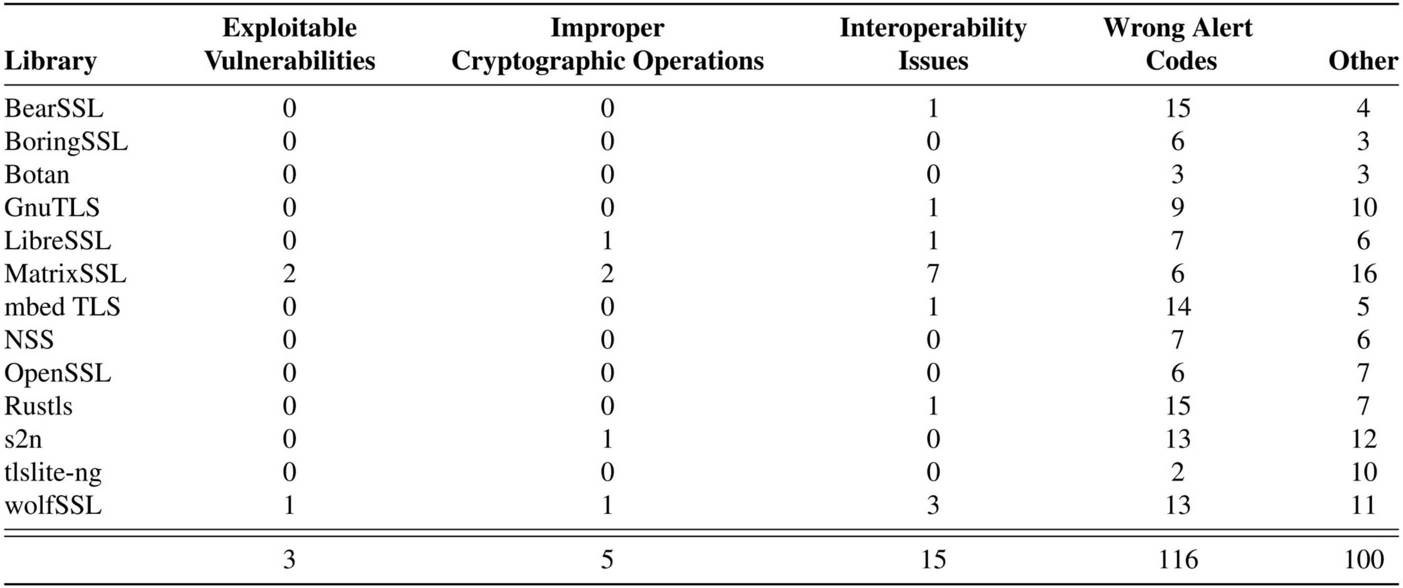
Library	Exploitable Vulnerabilities	Improper Cryptographic Operations	Interoperability Issues	Wrong Alert Codes
BearSSL	0	0	1	15
BoringSSL	0	0	0	6
Botan	0	0	0	3
GnuTLS	0	0	1	9
LibreSSL	0	1	1	7
MatrixSSL	2	2	7	6
mbed TLS	0	0	1	14
NSS	0	0	0	7
OpenSSL	0	0	0	6
Rustls	0	0	1	15
s2n	0	1	0	13
tlslite-ng	0	0	0	2
wolfSSL	1	1	3	13
	3	5	15	116

239 RFC violations



Library	Exploitable Vulnerabilities	Improper Cryptographic Operations	Interoperability Issues	Wrong Alert Codes	Other
BearSSL	0	0	1	15	4
BoringSSL	0	0	0	6	3
Botan	0	0	0	3	3
GnuTLS	0	0	1	9	10
LibreSSL	0	1	1	7	6
MatrixSSL	2	2	7	6	16
mbed TLS	0	0	1	14	5
NSS	0	0	0	7	6
OpenSSL	0	0	0	6	7
Rustls	0	0	1	15	7
s2n	0	1	0	13	12
tlslite-ng	0	0	0	2	10
wolfSSL	1	1	3	13	11
	3	5	15	116	100

239 RFC violations



Library	Exploitable Vulnerabilities	Improper Cryptographic Operations	Interoperability Issues	Wrong Alert Codes	Other
BearSSL	0	0	1	15	4
BoringSSL	0	0	0	6	3
Botan	0	0	0	3	3
GnuTLS	0	0	1	9	10
LibreSSL	0	1	1	7	6
MatrixSSL	2	2	7	6	16
mbed TLS	0	0	1	14	5
NSS	0	0	0	7	6
OpenSSL	0	0	0	6	7
Rustls	0	0	1	15	7
s2n	0	1	0	13	12
tlslite-ng	0	0	0	2	10
wolfSSL	1	1	3	13	11
	3	5	15	116	100

Overall, most libraries still passed a high percentage of tests

Conclusion

- TLS-Anvil, a test suite based on t-way testing
- 239 RFC violations found including 3 exploitable vulnerabilities
- Worth exploring for more RFCs and other protcols e.g QUIC

https://tls-anvil.com https://github.com/tls-attacker/TLS-Anvil @marcelmaehren