Improving the Security of Quantum Platforms using Combinatorial Methods

Workshop on Secure Protocol Implementations in the Quantum Era

Dimitris E. Simos[1,2], Manuel Leithner[2,3], Dominik Schreiber[3], Bernhard Garn[3]

[1]Paris Lodron University of Salzburg, Austria
[2]Salzburg University of Applied Sciences, Austria
[3]SBA Research, Austria

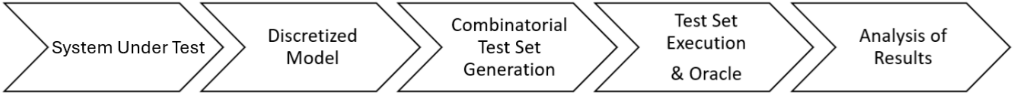
June 24, 2025
Munich, Germany

Outline of the Talk

Introduction to CT

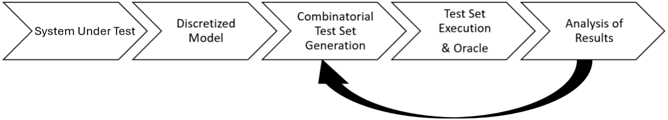
Combinatorial Testing (CT)

- Application of combinatorial methods to (software) testing problems
- *System under Test* modelled in terms of finitely many parameters taking finitely many values
- Test set generation based on combinatorial coverage criteria
- CT has been successfully applied to
  - Software configuration testing;
  - Software input data testing;
  - Hardware testing;
  - ML/AI testing;
  - Security testing.



Combinatorial Testing (CT)

- Application of combinatorial methods to (software) testing problems
- *System under Test* modelled in terms of finitely many parameters taking finitely many values
- Test set generation based on combinatorial coverage criteria
- CT has been successfully applied to
  - Software configuration testing;
  - Software input data testing;
  - Hardware testing;
  - ML/AI testing;
  - Security testing.
  - Test-cycle iterations / Combinatorial fault localization



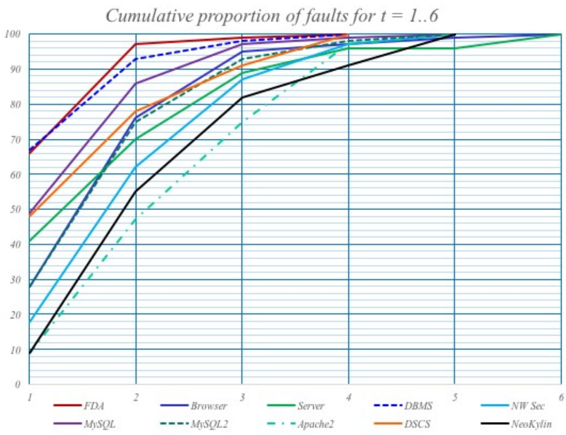


Figure: Visualization graph of results of NIST fault studies (NIST).

Implications for software testing

Observation from the NIST fault study:

- Most failures are induced by single factor faults;
- With progressively fewer failures induced by interactions between two or more factors.

Implications for software testing

Observation from the NIST fault study:

- Most failures are induced by single factor faults;
- With progressively fewer failures induced by interactions between two or more factors.

$\Longrightarrow$ Tests that cover all such few variable-interactions can be very effective!

Covering Arrays (structures in Discrete Mathematics)

Guaranteed diversity in terms of tuples

Definition

Let $t, k, v \in \mathbb{N} \wedge t \leq k$.
A covering array (CA) for the configuration $C = (t, k, v)$ is defined as

- an $N \times k$ array over a finite alphabet A of cardinality v,
- such that in any $N \times t$ subarray all possible t-tuples arising from this alphabet A appear at least once as rows in the selected subarray.

Denoted as $CA(N; t, k, v)$.

Covering Array Example

From combinatorial designs to sofware artifacts

- t-way test sets (i.e., combinatorial) derived from covering arrays
- every row is a test case; appropriate translation of values, below: 0 $\rightarrow$ False, 1 $\rightarrow$ True

0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	1	1	1
0	1	0	1	0	1	0	1	0	1
0	1	1	0	1	0	1	0	1	0
1	0	0	1	1	0	0	1	0	1
1	0	1	0	0	1	1	0	1	0
1	1	0	0	1	1	0	0	1	0
1	1	1	1	0	0	1	1	0	1
0	0	0	0	1	1	1	0	0	1
0	0	1	1	0	0	0	1	1	0
1	1	0	0	0	0	1	1	1	1
1	1	1	1	1	1	0	0	0	0
0	0	0	1	0	0	1	0	1	1
0	0	1	0	1	1	0	1	0	0
1	1	1	1	0	0	0	0	1	1
1	1	0	0	1	1	1	1	0	0
1	1	0	1	0	0	1	1	0	0
1	1	1	0	1	1	0	0	0	1

Figure: Binary strength three covering array with three highlighted 3-way column selections $(\{1,2,3\},\{4,5,7\},\{8,9,10\})$.

Definition

Let S be a nonempty set with $|S| = s \in \mathbb{N}^\times$ and $N, t \in \mathbb{N}^\times$ with $0 < t \leq s$.
Then, a sequence covering array $SCA(N, S, t)$ (*SCA*) of strength t is

- an $N \times s$ matrix,
- with entries from a finite set S,
- such that every t-way permutation of symbols from S occurs in at least one row (not necessarily adjacent), and
- each row is a permutation of the s symbols.

Sequence Covering Array Example

From combinatorial designs to sofware artifacts

- Sequence test sets derived from sequence covering arrays
- every row is a test case; appropriate translation of values, for example: 1 → "Mouse left-click on button", 2 → "Switch of window-focus"

Test	Sequences					
1	1	2	3	4	5	6
2	6	5	4	3	2	1
3	4	5	6	1	2	3
4	3	2	1	6	5	4
5	2	6	1	4	3	5
6	5	3	4	1	6	2
7	1	5	6	3	2	4
8	4	2	3	6	5	1
9	3	5	1	4	2	6
10	6	2	4	1	5	3

Figure: Sequence covering array for 6 symbols of strength three with 10 rows.

Combinatorial Design structures

Properties and observations

- Combinatorial design structures arise in discrete mathematics; exhibit many connections to coding- and graph theory
- Actual design construction is an empirically observed hard combinatorial optimization problem
- Different methods proposed for their construction, including (meta-) heuristics, combinatorial- and exact algorithms
- Designs with coverage properties are often significantly smaller than corresponding "full space":
  - CA(**18**;3,10,2) vs $2^{10} = 1024$, reduction of $\approx 98.24\%$
  - SCA(**6**,$\{1, 2, 3, 4, 5, 6\}$,3) vs $6! = 720$, reduction of $\approx 98.6\%$

References on CT

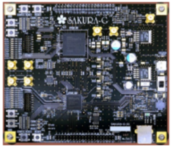
- Practical Combinatorial Testing; Richard Kuhn (NIST), Raghu Kacker (NIST), Yu Lei (NIST); NIST SP 800-142; `https://doi.org/10.6028/NIST.SP.800-142`
- R. N. Kacker, D. R. Kuhn, Y. Lei and D. E. Simos, "Measuring the Adequacy of a Test Suite With Respect to a Modeled Test Space," in IEEE Software, vol. 39, no. 5, pp. 62-67, Sept.-Oct. 2022, doi: 10.1109/MS.2021.3108060.
- R. Kuhn, R. N. Kacker, Y. Lei and D. Simos, "Input Space Coverage Matters," in Computer, vol. 53, no. 1, pp. 37-44, Jan. 2020, doi: 10.1109/MC.2019.2951980.
- Kacker, R.N., Kuhn, D.R., Lei, Y. et al. Factorials Experiments, Covering Arrays, and Combinatorial Testing. Math.Comput.Sci. 15, 715–739 (2021). https://doi.org/10.1007/s11786-021-00502-7
- M. S. Raunak, D. R. Kuhn, R. N. Kacker and J. Y. Lei, "Combinatorial Testing for Building Reliable Systems," in IEEE Reliability Magazine, vol. 1, no. 1, pp. 15-19, March 2024
- M. S. Raunak, D. R. Kuhn, R. N. Kacker and Y. Lei, "Ensuring Reliability Through Combinatorial Coverage Measures," in IEEE Reliability Magazine, vol. 1, no. 2, pp. 20-26, June 2024, doi: 10.1109/MRL.2024.3389629

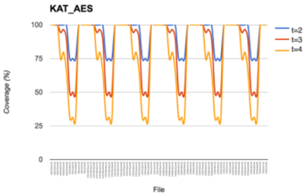
CST for TLS

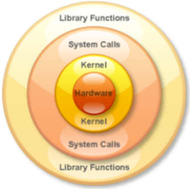
Large scale automated software testing for security

- Complex web applications
- Linux kernels
- Protocol testing & crypto alg. validation
- Hardware Trojan horse (HTH) detection

Combinatorial methods can make **software security testing** much more **efficient** and effective than conventional approaches

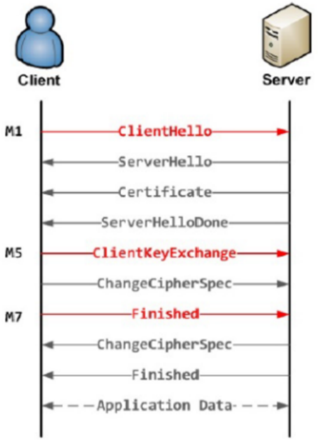






Combinatorial methods for TLS testing

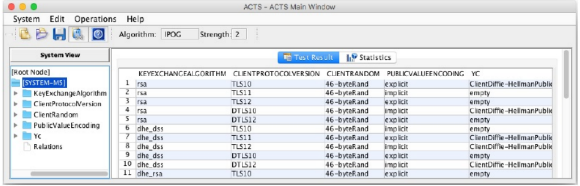
- **Input Test Space for CT:** Employ Input Parameter Modelling (IPM)
- **TLS Specification:** Select parameters and possible values for M1, M5 and M7
- Three different models are constructed which give rise to three distinctive test sets according to standard



Input models for TLS messages

M5:

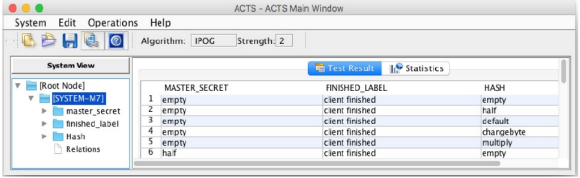
```
KeyExchangeAlgorithm : rsa,
dhe_dss, dhe_rsa, dh_dss,
dh_rsa, dh_anon
ClientProtocolVersion :
TLS10, TLS11, TLS12, DTLS10,
DTLS12
ClientRandom : 46-byteRand
PublicValueEncoding :
implicit, explicit
Yc : empty, ClientDiffie -
HellmanPublicValue
```



	KEYEXCHANGEALGORITHM	CLIENTPROTOCOLVERSION	CLIENTRANDOM	PUBLICVALUEENCODING	YC
1	rsa	TLS10	46-byteRand	explicit	ClientDiffie-HellmanPublic
2	rsa	TLS11	46-byteRand	implicit	empty
3	rsa	TLS12	46-byteRand	explicit	empty
4	rsa	DTLS10	46-byteRand	implicit	ClientDiffie-HellmanPublic
5	rsa	DTLS12	46-byteRand	explicit	empty
6	dhe_dss	TLS10	46-byteRand	implicit	empty
7	dhe_dss	TLS11	46-byteRand	explicit	ClientDiffie-HellmanPublic
8	dhe_dss	TLS12	46-byteRand	implicit	ClientDiffie-HellmanPublic
9	dhe_dss	DTLS10	46-byteRand	explicit	empty
10	dhe_dss	DTLS12	46-byteRand	implicit	ClientDiffie-HellmanPublic
11	dhe_rsa	TLS10	46-byteRand	explicit	empty

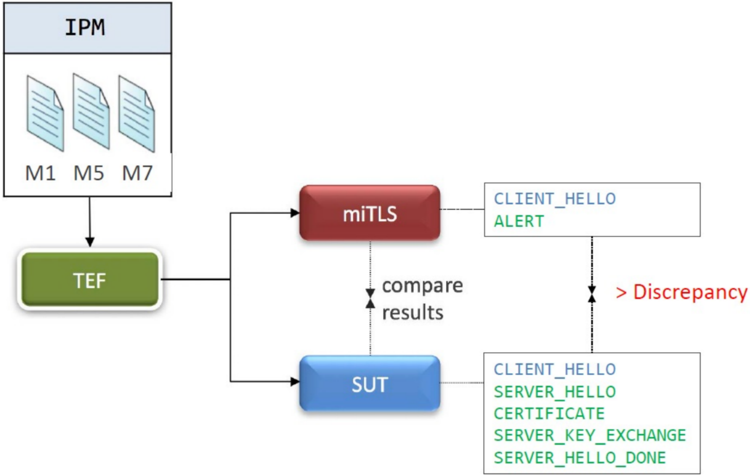
M7:

```
master_secret : empty, half,
default, changebyte, multiply
finished_label : client
finished
Hash : empty, half, default,
changebyte, multiply
```



	MASTER_SECRET	FINISHED_LABEL	HASH
1	empty	client finished	empty
2	empty	client finished	half
3	empty	client finished	default
4	empty	client finished	changebyte
5	empty	client finished	multiply
6	half	client finished	empty

Test execution framework for TLS testing



- Previous TLS versions
  - SSLv{2,3.0} (v3.0 RFC 6101, deprecated by RFC 7568)
  - TLS 1.0 (RFC 2246, deprecated by RFC 8996)
  - TLS 1.1 (RFC 4346, deprecated by RFC 8996)
  - TLS 1.2 (RFC 5246)
- TLS 1.3 (most recent)
  - RFC 8446, August 2018
  - 48.09% supported of Alexa1M (Dec 31, 2020)
  - Improved handshake
  - Stricter ciphers
- Many TLS security issues in the past:
  - Protocol issues (including POODLE, DROWN)
  - Implementation issues (including RACOON, HeartBleed)
  - Policy-related issues
- NO PQC-aspects officially in TLS available yet!

Transitioning to PQC-TLS

New situation after NIST standardization rounds completed

- NIST standardization process "completed" (as of June 24, 2025):
  - FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard;
  - FIPS 204: Module-Lattice-Based Digital Signature Standard;
  - FIPS 205: Stateless Hash-Based Digital Signature Standard;
  - NIST IR 8545: HQC (Hamming Quasi-Cyclic, selected for standardization)
- Native PQC-TLS
  - Use only (NIST-standardized) PQC-schemes directly in handshake
- Parallel classic-PQC TLS (towards PQC-transition)
  - More choices for selective use in parallel
- Hybrid classic-PQC TLS (towards PQC-transition)
  - Combine classic and PQC-schemes in order to take advantage of both
- Common implementation efforts for PQC-primitives
  - Open Quantum Safe (OQS) project (among others)
- Recommendations for cipher suites (including Suite B, CNSA 1.0/2.0)

References on CST for TLS testing

- D. E. Simos, R. Kuhn, A. G. Voyiatzis and R. Kacker, "Combinatorial Methods in Security Testing," in Computer, vol. 49, no. 10, pp. 80-83, Oct. 2016, doi: 10.1109/MC.2016.314
- J. Bozic, K. Kleine, D. E. Simos and F. Wotawa, "Planning-Based Security Testing of the SSL/TLS Protocol," 2017 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), Tokyo, Japan, 2017, pp. 347-355, doi: 10.1109/ICSTW.2017.63.
- Simos, D.E., Bozic, J., Garn, B. et al. Testing TLS using planning-based combinatorial methods and execution framework. Software Qual J 27, 703–729 (2019). `https://doi.org/10.1007/s11219-018-9412-z`
- B. Garn, D. E. Simos, F. Duan, Y. Lei, J. Bozic and F. Wotawa, "Weighted Combinatorial Sequence Testing for the TLS Protocol," 2019 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), Xi'an, China, 2019, pp. 46-51, doi: 10.1109/ICSTW.2019.00031.

CT for X.509 certificates

Structure Modelling of X.509 for CT

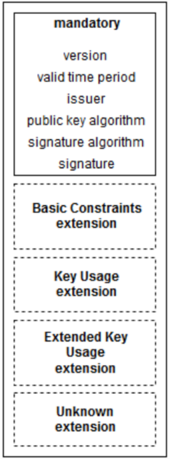


Figure: Blocks in an X.509 certificate.

```
Version(5)::= -1 | 0 | 1 | 2 | 3
Validity_Time(4)::= expired | not_yet_valid |
    start_future_end_past | valid
Issuer(3)::= Chain | Self | Unrelated
Key_Type(2)::= RSA | DSA
Signature_Type(3)::= Chain | Self | Unrelated
Signature_Algorithm(3)::= MD5 | SHA1 | SHA256
Ext_Basic_Constraints_enabled(2)::= 0 | 1
Ext_Basic_Constraints_critical(3)::= 0 | 1 |
    not_active
Ext_Basic_Constraints_CA(3)::= 0 | 1 |
    not_active
Ext_Basic_Constraints_pathlen(4)::= -1 | 0 | 1
     | not_active
Ext_Keyusage_enabled(2)::= 0 | 1
Ext_Keyusage_critical(3)::= 0 | 1 | not_active
Ext_Extended_Keyusage_enabled(2)::= 0 | 1
Ext_Extended_Keyusage_critical(3)::= 0 | 1 |
    not_active
Ext_unknown_enabled(2)::= 0 | 1
Ext_unknown_critical(3)::= 0 | 1 | not_active
```

Figure: IPM for X.509 certificate for CT.

Certificate generation with CT

Guaranteed combinatorial coverage of the input space

- Combinatorial sampling strategies:
  - *Intra-block*
  - *Inter-block*
  - *flat*

Mandatory Block				Basic Constraint Extension Block			
version	hash	key	signature	active	critical	is_authority	pathlen
0	md5	dsa	self	true	false	false	1
0	sha1	rsa	unrelated	false	dummy	dummy	dummy
0	sha256	dsa	parent	true	true	true	0
1	md5	rsa	unrelated	true	true	false	0
1	sha1	rsa	parent	true	false	true	1
1	sha256	dsa	self	false	dummy	dummy	dummy
2	md5	rsa	parent	false	dummy	dummy	dummy
2	sha1	dsa	self	true	true	true	0
2	sha256	rsa	unrelated	true	false	false	1
1	md5	dsa	unrelated	true	false	true	0
2	sha1	dsa	parent	true	true	false	1
0	sha256	rsa	self	false	dummy	dummy	dummy

Figure: Pairwise (i.e., 2-way) abstract test set for simplified certificate model.

t	Intra	Inter	Flat
2	20	28	26
3	73	107	126
4	210	372	536
5	551	1,110	1,965
6	1,020	2,709	6,598
7	1,020	4,904	20,487

Figure: Strength vs generated number of certificates.

Test execution results and comparison

Error	BouncyCastle	wolfSSL	GnuTLS	NSS	OpenJDK	OpenSSL	mbed
untrusted	✓	✓	✓	✓	✓	✓	✓
expired or not yet valid	✓	✓	✓	✓	✓	✓	✓
parse-error	✓	✓	✓	✓	✓	✗	✓
crash	✗	✓	✗	✗	✗	✗	✗
use of insecure algorithm	✗	✗	✓	✓	✗	✗	✓
invalid signature	✗	✓	✓	✓	✗	✗	✗
unknown critical extension	✗	✗	✗	✓	✗	✓	✗
extension in non-v3 cert	✗	✗	✗	✗	✓	✗	✗
use of weak key	✗	✗	✗	✗	✗	✗	✓
name constraint violation	✗	✗	✗	✓	✗	✗	✗
key usage not allowed	✗	✗	✗	✓	✗	✗	✗

Figure: Observed returned error tuples.

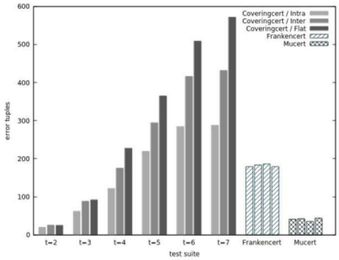


Figure: Number of different error tuples per test set.

Quo Vadis, X.509?

Status & Current and future challenges for X.509

- Widely used standard for PKC-based authentication (including RFC 5280)
- Currently standandized for classic cryptographic schemes only
  - 'draft-ietf-tls-hybrid-design-13', most recent update June 17, 2025: Hybrid key exchange in TLS 1.3
  - 'draft-ounsworth-pq-composite-sigs-13', most recent update March 4, 2024: Composite ML-DSA for use in Internet PKI
- Proposed transition approaches:
  - PQC-native
  - Classic/PQC-combined via extensions in certificates
  - Hybrid (i.e., composite algorithms) between classic and pqc
  - Classic and PQC independently in parallel deployed
- Certificate (chain) validation: Under which conditions will a presented certificate (chain) by accepted?
  - Root certificate
  - (Zero or more) Intermediate certificate(s)
  - Leaf certificate
- Transition-plath for certificate authorities:
  - Root certificates in Mozilla Root Store expire up to year 2046
  - New root certificates according to which (hybrid?) PQC-scheme?
  - How to distribute new root certificates in parallel to software (hardware?) support?

References on CT for X.509 certificate testing

SPIQE 2025

D. E. Simos
M. Leithner
D. Schreiber
B. Garn

Introduction to CT

CST for TLS

CT for X.509 certificates

Conclusion & Future Work

- K. Kleine and D. E. Simos, "Coveringcerts: Combinatorial Methods for X.509 Certificate Testing," 2017 IEEE International Conference on Software Testing, Verification and Validation (ICST), Tokyo, Japan, 2017, pp. 69-79, doi: 10.1109/ICST.2017.14.

Conclusion & Future Work

More (Testing) Challenges for PQC ahead!

- Transition has to start as soon as possible ("SNDL: store-now, decrypt later" attack)
- Hybrid classic-pqc schemes starting to be used in industry
  - META internally with implementation Fizz: Hybridization of Kyber with X25519
- Hybrid schemes evaluation for official FIPS-compliance?
- 75 billion IoT devices in use by 2025 (NCCOE/NIST)
  1. Already use lightweight cryptographic schemes
  2. Upgrade software/hardware to support PQC-TLS?
  3. Upgrade software/hardware to support PQC-X.509?
- How to test compatibility & security of more of billions of devices?
  - Native/hybrid approaches?
  - How to distribute "PQC-root certificates" in parallel with classic ones?
- How to react when flaw is discovered in one of the 4 (5?) PQC-schemes standardized by NIST?
  - In the context of native/hybrid PQC-TLS?
  - In the context of native/hybrid PQC-X.509?
- Impact of EO (June 6, 2025): "SUSTAINING SELECT EFFORTS TO STRENGTHEN THE NATION'S CYBERSECURITY AND AMENDING EXECUTIVE ORDER 13694 AND EXECUTIVE ORDER 14144"

Conclusion & Future Work

Conclusion

- Testing systems using PQC is critical!
- Combinatorial methods provide the means to obtain:
  - Guarantees of structural coverage of the input space;
  - Minimized test set sizes.

Conclusion & Future Work

Conclusion

- Testing systems using PQC is critical!
- Combinatorial methods provide the means to obtain:
  - Guarantees of structural coverage of the input space;
  - Minimized test set sizes.

Future Work

- Explore (some) of the C(S)T-ideas presented!

Thank you very much for your attention!

Questions?

Contact information:

BGarn@sba-research.org
dimitrios.simos@plus.ac.at