Combinatorial Coverage Analysis of Subsets of the TLS Cipher Suite Registry

Dimitris E. Simos
SBA Research
A-1040 Vienna, Austria
dsimos@sba-research.org

Kristoffer Kleine
SBA Research
A-1040 Vienna, Austria
kkleine@sba-research.org

Rick Kuhn
Computer Security Division
NIST
Gaithersburg, MD, USA
d.kuhn@nist.gov

Raghu Kacker
Applied and Computational
Mathematics Division, NIST
Gaithersburg, MD, USA
raghu.kacker@nist.gov

***Abstract*—We present a combinatorial coverage measurement analysis for (subsets) of the TLS cipher suite registries by analyzing the specified ciphers of IANA, ENISA, BSI, Mozilla and NSA Suite B. The method introduced here may contribute towards the design of quality measures of cipher suites, and may also be applied more broadly to the analysis of configurable systems.**

Keywords—Combinatorial testing, coverage, measurement, TLS, subsets, cipher suites.

I. Introduction

Security protocols continue to suffer from security flaws in their implementations, like the POODLE attack in SSL/TLS or the Heartbleed bug in the OpenSSL cryptographic library. Clearly, additional steps have to be taken to ensure or better contribute towards their quality assurance, as part of the testing cycle where critical points in the system state-space are covered. The full system state-space, consisting of all valid configurations, is generally impossible to cover, because the number of configurations is too large. However, empirical research shows that the number of factors interacting in system failures is relatively small [1]. This has also been confirmed in the case of web application security testing [2].

II. Combinatorial Coverage

Empirical data show that a significant number of software failures are induced by the interaction of two or more factors, and interaction faults can be extremely difficult to identify. Thus it is useful to measure the proportion of 2-way, 3-way, and higher strength combinations that are covered by a test set. Any combinations that have not been tested represent a portion of the input space for which the application has not been shown to be correct. Measuring the proportion of the input space for which the system response is untested and unknown can thus provide a useful quantity in estimating residual risk after testing. We explain the concept of combinatorial coverage measurement, a variety of measures that are available, and theorems relating (static) combinatorial coverage to (dynamic) structural coverage. These concepts are illustrated with examples comparing measures of tests for a NASA spacecraft and open source test configurations for the TLS cipher suite, which is the main focus of this paper.

A configuration with n variables contains $\binom{n}{t}$ t-way combinations, so a test set with many configurations will contain a large number of combinations. *Combinatorial coverage* measures the inclusions of t-way combinations in a test set. Note that this measure is different from conventional structural coverage metrics (such as statement or branch coverage) and is independent of these other measures. Because combinatorial coverage measures the input space that is tested, and consequently also the untested portion of input space, it is a useful in gauging the residual risk after testing. A variety of combinatorial coverage measures are available, including a fundamental measure of *total variable-value configuration coverage*: for a given combination of t variables, the proportion of all t-way value settings that are covered by at least one test case in a test set [3].

For example, two binary variables have four possible settings. Consider four tests containing variables a, b, c, and d: $\{0000, 0110, 1001, 0111\}$. There are $\binom{4}{2} = 6$ possible variable combinations and $2^2 \times \binom{4}{2} = 24$ possible variable-value configurations. Of these, 19 variable-value configurations are covered and the only ones missing are $ab = 11$, $ac = 11$, $ad = 10$, $bc = 01$, $bc = 10$, so the total variable-value configuration coverage is 19/24 = 79%. These measures are shown in Figure 1, where the upper right-hand corner represents the 21% of the 2-way combinations in the input space not tested. Figure 2 shows measurements for 2-way through 5-way combination coverage for 7,489 tests for a NASA spacecraft. Note that the untested portion for 2-way combinations (above red line) is only about 6% of the total, and 3-way to 5-way coverage is relatively high. In contrast, as we shall see shortly after the situation changes rapidly when measuring the combination coverage for the TLS cipher suites.

III. Input Models for Cipher suites

A cipher suite is a combination of key exchange, authentication, encryption and MAC algorithms which are used together to provide the security of TLS. For example, the cipher suite `TLS_RSA_WITH_AES_256_GCM_SHA384` specifies that session secrets are exchanged using RSA while AES with a 256 bit key is used for encrypting the application data and integrity is provided by SHA384. These combinations are specified in various RFCs. For example, NSA Suite B (before a 2015 revision) consisted of the 2 cipher suites `TLS_ECDHE_ECDSA_AES_128_GCM_SHA256` and `TLS_ECDHE_ECDSA_AES_256_GCM_SHA384`. We have developed an input parameter model (IPM) for splitting the suites into parameters (Table XI). It is revealed that 2-way

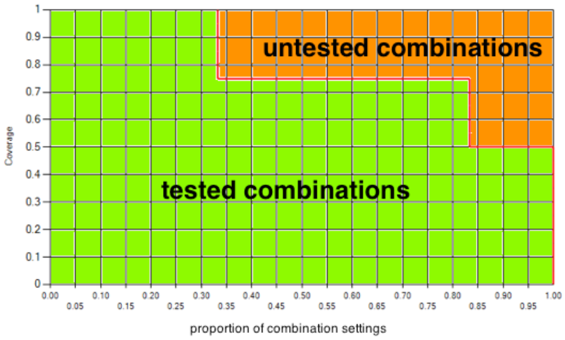

Fig. 1: Example test set

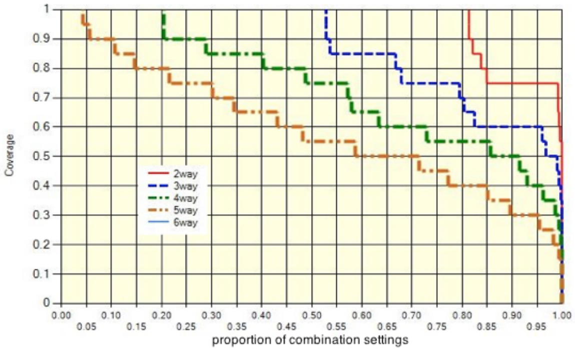

Fig. 2: Measured combinatorial coverage for 7,489 tests.

coverage is achieved for all pairs of parameters except for the parameter pair (Key size, MAC) where the tuples (256, SHA256) and (128, SHA384) are missing.

In addition, we have performed a comparison of the specified cipher suites of IANA, ENISA, BSI, Mozilla and NSA Suite B and our measurement results are given in Table XII. Note that TLS is used only as an illustration of the analysis method, because complex constraints embedded in the TLS code of different implementations have not been included.

For example, if encryption is selected as `NULL` in the IANA cipher suites, then the key length must be zero and the mode must be `NULL` as well. Similarly, when the AES key size is 128 bits in NSA suite B, then the hash function must be SHA-256 with 128-bit collision resistance to match the security strength; and when different curves are used with `ECDHE_ECDSA`, key lengths must be changed. The figures in Table XII can therefore be considered *upper bounds rather than exact sizes of the configuration spaces*. A complete analysis including specific TLS constraints can be considered in a future paper.

A. IANA

The Internet Assigned Numbers Authority (IANA) records all cipher suites which have been specified for TLS (versions 1.0, 1.1 and 1.2) and each cipher suite is assigned a unique identifier (2-byte value).[1] The whole cipher suite list contains 317 cipher suites which are omitted for space reasons, but we give the resulting IPM in Table I. In the context of this paper, we consider a cipher suite list as a test set.

KEX	Enc	Key size	Mode	Hash
NULL	NULL	0	NULL	NULL
RSA	RC4	40	CBC	MD5
RSA_EXPORT	RC2	56	EDE_CBC	SHA
DH_DSS_EXPORT	IDEA	128	GCM	SHA256
DH_DSS	DES	168	CCM	SHA384
DH_RSA_EXPORT	3DES	256	CCM_8	
DH_RSA	AES			
DHE_DSS_EXPORT	CAMELLIA			
DHE_DSS	SEED			
DHE_RSA_EXPORT	ARIA			
DHE_RSA				
DH_anon_EXPORT				
DH_anon				
KRB5				
KRB5_EXPORT				
PSK				
DHE_PSK				
RSA_PSK				
ECDH_ECDSA				
ECDHE_ECDSA				
ECDH_RSA				
ECDHE_RSA				
ECDH_anon				
SRP_SHA				
SRP_SHA_RSA				
SRP_SHA_DSS				
ECDHE_PSK				
PSK_DHE				

TABLE I: IANA IPM

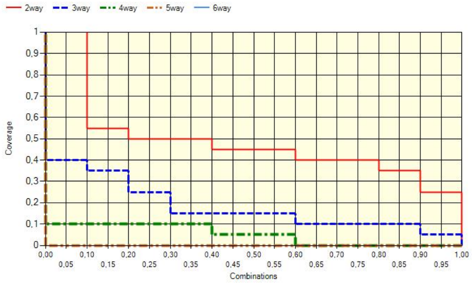

Fig. 3: Coverage IANA

1) Key length constraints:

- For each encryption algorithm one constraint for allowed key sizes (e.g. AES ⇒ key size = 128 or 256)
- Only necessary for IANA model since the other subsets don't allow for invalid combinations

B. ENISA

The following recommendation is issued by the ENISA (European Union Agency for Network and Information Security). The recommendation notes that none of the available key exchange mechanisms are particularly favorable for future use (long term) as no proof of security exists, but recommends (EC)DHE together with RSA, DSS or ECDSA for legacy use as this provides forward secrecy [2]. See Table II for the whole list of cipher suites.

[1] https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml#tls-parameters-4

[2] https://www.enisa.europa.eu/activities/identity-and-trust/library/deliverables/study-on-cryptographic-protocols/at%5fdownload/fullReport

KEX	Enc	Key size	Mode	Hash
DHE_DSS	CAMELLIA	128	GCM	SHA256
DHE_DSS	AES	128	GCM	SHA256
DHE_DSS	CAMELLIA	256	GCM	SHA384
DHE_DSS	AES	256	GCM	SHA384
DHE_RSA	CAMELLIA	128	GCM	SHA256
ECDHE_RSA	CAMELLIA	128	GCM	SHA256
DHE_RSA	AES	128	GCM	SHA256
ECDHE_RSA	AES	128	GCM	SHA256
DHE_RSA	CAMELLIA	256	GCM	SHA384
ECDHE_RSA	CAMELLIA	256	GCM	SHA384
DHE_RSA	AES	256	GCM	SHA384
ECDHE_RSA	AES	256	GCM	SHA384
DHE_RSA	AES	128	CCM	SHA256
DHE_RSA	AES	128	CCM_8	SHA256
DHE_RSA	AES	256	CCM	SHA256
DHE_RSA	AES	256	CCM_8	SHA256
ECDHE_ECDSA	CAMELLIA	128	GCM	SHA256
ECDHE_ECDSA	AES	128	GCM	SHA256
ECDHE_ECDSA	CAMELLIA	256	GCM	SHA384
ECDHE_ECDSA	AES	256	GCM	SHA384
ECDHE_ECDSA	AES	128	CCM	SHA256
ECDHE_ECDSA	AES	128	CCM_8	SHA256
ECDHE_ECDSA	AES	256	CCM	SHA256
ECDHE_ECDSA	AES	256	CCM_8	SHA256

TABLE II: ENISA recommended cipher suites

KEX	Enc	Key size	Mode	Hash
ECDHE_ECDSA	AES	128	GCM	SHA256
ECDHE_RSA	CAMELLIA	256	CCM	SHA384
DHE_RSA			CCM_8	
DHE_DSS				

TABLE III: ENISA IPM

C. BSI

The BSI, a German Federal Agency responsible for computer and network security, gives out recommendations for cipher suites it considers secure to use. See table IV for a full list of these cipher suites. [3]

D. Mozilla

Mozilla recommends specific cipher suites for server side TLS as a guideline for helping system administrators harden the configuration of servers, most notably webservers [4]. We analysed the recommended cipher list for modern compatibility. See table VI for a full list of cipher suites.

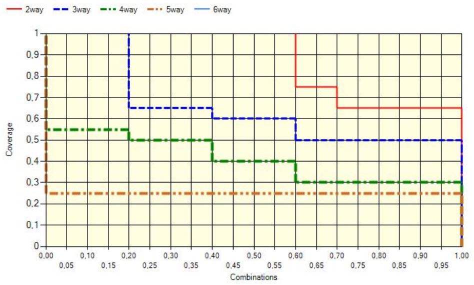

Fig. 4: Coverage ENISA

KEX	Enc	Key size	Mode	Hash
ECDHE_ECDSA	AES	128	CBC	SHA256
ECDHE_ECDSA	AES	256	CBC	SHA384
ECDHE_ECDSA	AES	128	GCM	SHA256
ECDHE_ECDSA	AES	256	GCM	SHA384
ECDHE_RSA	AES	128	CBC	SHA256
ECDHE_RSA	AES	256	CBC	SHA384
ECDHE_RSA	AES	128	GCM	SHA256
ECDHE_RSA	AES	256	GCM	SHA384
DHE_DSS	AES	128	CBC	SHA256
DHE_DSS	AES	256	CBC	SHA256
DHE_DSS	AES	128	GCM	SHA256
DHE_DSS	AES	256	GCM	SHA384
DHE_RSA	AES	128	CBC	SHA256
DHE_RSA	AES	256	CBC	SHA256
DHE_RSA	AES	128	GCM	SHA256
DHE_RSA	AES	256	GCM	SHA384

TABLE IV: BSI recommended cipher suites

KEX	Enc	Key size	Mode	Hash
ECDHE_ECDSA	AES	128	CBC	SHA256
ECDHE_RSA		256	GCM	SHA384
DHE_RSA				
DHE_DSS				

TABLE V: BSI IPM

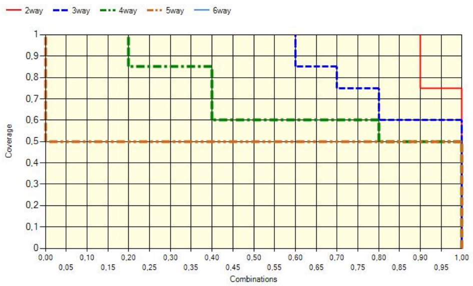

Fig. 5: Coverage BSI

KEX	Enc	Key size	Mode	Hash
ECDHE_ECDSA	AES	128	CBC	SHA
ECDHE_ECDSA	AES	256	CBC	SHA
ECDHE_RSA	AES	128	CBC	SHA
ECDHE_RSA	AES	256	CBC	SHA
ECDHE_ECDSA	AES	128	CBC	SHA256
ECDHE_ECDSA	AES	256	CBC	SHA384
ECDHE_RSA	AES	128	CBC	SHA256
ECDHE_RSA	AES	256	CBC	SHA384
ECDHE_ECDSA	AES	128	GCM	SHA256
ECDHE_ECDSA	AES	256	GCM	SHA384
ECDHE_RSA	AES	128	GCM	SHA256
ECDHE_RSA	AES	256	GCM	SHA384
DHE_RSA	AES	128	CBC	SHA
DHE_DSS	AES	256	CBC	SHA
DHE_RSA	AES	256	CBC	SHA
DHE_DSS	AES	128	CBC	SHA256
DHE_RSA	AES	128	CBC	SHA256
DHE_RSA	AES	256	CBC	SHA256
DHE_RSA	AES	128	GCM	SHA256
DHE_RSA	AES	256	GCM	SHA384
DHE_DSS	AES	128	GCM	SHA256
DHE_DSS	AES	256	GCM	SHA384

TABLE VI: Mozilla recommended cipher suites

[3] https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102-2.pdf?%5f%5fblob=publicationFile&v=1

[4] https://wiki.mozilla.org/Security/Server_Side_TLS

KEX	Enc	Key size	Mode	Hash
ECDHE_ECDSA ECDHE_RSA DHE_RSA DHE_DSS	AES	128 256	CBC GCM	SHA SHA256 SHA384

TABLE VII: Mozilla IPM

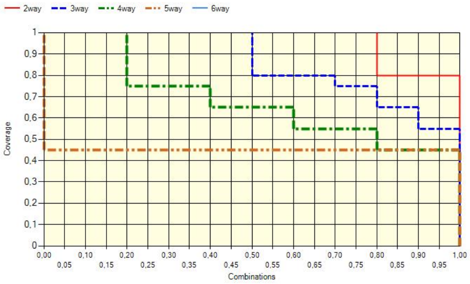

Fig. 6: Coverage Mozilla

KEX	Enc	Key size	Mode	Hash
ECDHE_ECDSA	AES	128 256	GCM	SHA256 SHA384

TABLE X: NSA IPM (before revision)

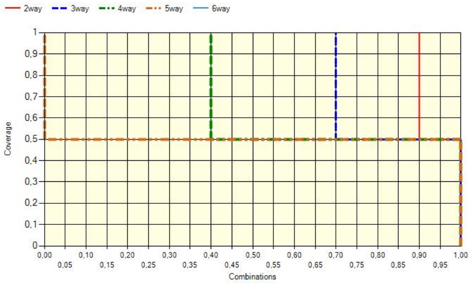

Fig. 7: Coverage NSA Suite B

E. NSA Suite B

Suite B is a recommendation by the NSA [5]. Currently only one cipher, namely `TLS_ECDHE_ECDSA_AES_256_GCM_SHA384`, is recommended. Before a revision in 2015 AES 128 and SHA256 were also allowed.

IV. Measuring TLS Cipher Suites

The TLS cipher suites can be viewed as a collection of configuration settings or options, conditioned that an input parameter model is available. A particular implementation is composed from a number of modules or components that together provide desired functionality. For TLS, the components are of the five types of modules described earlier in Section III. Combination coverage is of interest for configurable systems because interactions between multiple components are often the source of bugs and vulnerabilities. The more potential interactions, the greater the possibility for such interoperability problems, and thus the greater need for testing. The significance of t-way combinations of configuration options is dependent on the application. For TLS cipher suites, an example might be the importance of analyzing the existing pairs of encryption and authentication functions. If encryption is provided without authentication, or with inadequately secure authentication, then users will be vulnerable to a man-in-the-middle attack.

If we wish to analyze a cipher suite, one consideration is the extent to which we can measure combinations of its configurable options. If a new or a revised cipher suite is proposed, for example, interoperability errors may be more common where combinations of options have not been used in the previous versions. For instance, if such a cipher suite has a pair of components that is present in the current suite, then it is already in use and interoperability problems are presumably more likely to have been identified through use. If a particular combination of components was not present in the current implementation, then, in the case that a testing procedure can be applied, this is likely to be required to ensure correct operation than if the current suite already has this pair. Furthermore, if an existing configuration uses both components, then previous test sets should have covered this pair. By identifying pairs and higher strength t-way combinations that are not covered in the current test set, we can improve the test sets by covering the previously untested interactions.

Consider Table II for example. The ENISA cipher suite input model has a configuration of $2^3 3^1 4^1$, for 96 possible implementations. Table XII contains 24 rows, so many other implementations are possible using software for each of the parameters in the input model. Table XII shows that 86% of the pairs have been covered in the ENISA specification, so problems that are related to unspecified 2-way interactions are relatively unlikely if a new option is afterwards added to a revised ENISA cipher suite.

The IANA cipher suite list, on the other hand, has an enormous possible configuration space, with an input model of $5^1 6^2 10^1 28^1$ = 8400 possible implementations. As shown in Table XII, only 45% of the pairs, and only 16% of 3-way combinations are present in the current list. Thus changes or additions are more likely to introduce combinations that have not been used in the existing test sets.

All data can be found in Table XII and is further visualized in Figures 3, 4, 5, 6 and 7 which show the coverage for the IANA, ENISA, BSI, Mozilla and NSA test sets. These figures

KEX	Enc	Key size	Mode	Hash
ECDHE_ECDSA	AES	128	GCM	SHA256
ECDHE_ECDSA	AES	256	GCM	SHA384

TABLE VIII: NSA recommended cipher suites before 2015 revision

KEX	Enc	Key size	Mode	Hash
ECDHE_ECDSA	AES	256	GCM	SHA384

TABLE IX: NSA recommended cipher suite after 2015 revision

[5] https://tools.ietf.org/html/rfc6460

Key exchange	Enc	Key size	Mode	MAC
ECDHE_ECDSA	AES	128	GCM	SHA256
ECDHE_ECDSA	AES	256	GCM	SHA384

TABLE XI: Test set for NSA Suite B

t	IANA	ENISA	BSI	Mozilla	NSA
2	45.55%	86.36%	97.83%	96.36%	89.47%
3	15.51%	65.24%	86.96%	82.5%	76.00%
4	5.41%	43.0%	69.32%	63.71%	62.50%
5	2.52%	25.0%	50.00%	45.83%	50.00%

TABLE XII: Combinatorial coverage ($t \in \{2, 3, 4, 5\}$)

show the proportion of combinations which are covered to a certain extent. For example, in Figure 6 we can see that 50% of all 3-way parameter combinations are fully covered while 70% are covered with at least 80%, and so on.

As noted previously, increasing the number of potential interactions between components may also increase the risk of bugs or vulnerabilities arising from feature interactions. Notice that the NSA Suite B specification contained only two configurations in the past, and now only one, thus limiting the potential for unknown interactions.

V. Cross coverage

We can consider the idea of cross coverage, where coverage is computed for one test array, A, using an input model, M', for a different array, A' of the same kind. That is, the measures produced give the coverage of M' by the tests in A. To the best of our knowledge, the idea of cross coverage is new and it has not been investigated elsewhere. Properties associated with this construct are topics for future papers, but we can review some implications with respect to applications, using the figures shown in Table XII and Table XIII.

Table XII shows the coverage of the five different input models in the header line by their respective cipher suites. For example, the IANA cipher suite covers 46% of the potential 2-way interactions among its components. As noted earlier, TLS is being used here only as an illustration of the analysis method, and constraints (on the related input models) have not been included in the measurement.

In Table XIII, the other four suites are measured in their coverage of the IANA input model. Thus the potential interactions among components of the ENISA suite are 5.8%. Of all the 2-way interactions that could be constructed among the components of the IANA input model, the IANA cipher suite covers 46%, and the ENISA suite covers 5.8%. If potential interactions are a source of problems, and thus represent a need for testing, then we can infer that less testing will be required when constructing an ENISA test set from the individual components for encryption algorithm, mode, etc. than for constructing an IANA test set.

VI. Conclusion and Future Work

In this work, we presented an analysis of the combinatorial coverage of subsets of the TLS cipher suite registry. This analysis was made feasible with the aid of an input model we developed for this cause. Our measurement results (with respect to the input model) indicate that there is a vast number of uncovered configurations in the specified cipher suites for TLS. However, complex TLS code constraints have not been included in this measurement and hence our results should be interpreted as upper bounds rather than exact sizes of the specific configuration spaces.

t	ENISA	BSI	Mozilla	NSA
2	5.83%	4.60%	5.42%	1.74%
3	1.57%	1.17%	1.49%	0.28%
4	0.46%	0.33%	0.43%	0.05%
5	0.19%	0.13%	0.17%	0.02%

TABLE XIII: Cross coverage with IANA IPM ($t \in \{2, 3, 4, 5\}$)

Whether we can view these measurement results as an indication for the root cause of security vulnerabilities is an important research topic that needs to be addressed further. In future work, we plan to undertake measures towards this direction by examining the relation of uncovered configurations and "weak" cipher suites, in the sense of the security strength these ciphers provide, in real data sets for TLS. Within this line of research, it would be also interesting to simulate a similar analysis for cipher suite lists that have been deprecated in newer TLS versions.

Acknowledgment

The research presented in the paper has been funded in part by the Austrian Research Promotion Agency (FFG) under grant 851205 (Security ProtocoL Interaction Testing in Practice - SPLIT) and the Austrian COMET Program (FFG).

Disclaimer: *Products may be identified in this document, but identification does not imply recommendation or endorsement by NIST, nor that the products identified are necessarily the best available for the purpose.*

References

[1] D. R. Kuhn, D. R. Wallace, and A. M. Gallo, Jr., "Software fault interactions and implications for software testing," *IEEE Trans. Softw. Eng.*, vol. 30, no. 6, pp. 418–421, Jun. 2004.

[2] B. Garn, I. Kapsalis, D. E. Simos, and S. Winkler, "On the applicability of combinatorial testing to web application security testing: A case study," in *Proceedings of the 2nd International Workshop on Joining AcadeMiA and Industry Contributions to Testing Automation (JAMAICA'14)*. ACM, 2014.

[3] D. Kuhn, I. Dominguez Mendoza, R. Kacker, and Y. Lei, "Combinatorial coverage measurement concepts and applications," in *Software Testing, Verification and Validation Workshops (ICSTW), 2013 IEEE Sixth International Conference on*, March 2013, pp. 352–361.