A Taxonomy of Metrics for Cryptographic Systems

Kimmo Halunen*, Mikko Kiviharju†, Jani Suomalainen*,
Visa Vallivaara*, Markku Kylänpää* and Outi-Marja Latvala*
*VTT Technical Research Centre of Finland Ltd
Espoo, Finland
email: firstname.lastname@vtt.fi
†Finnish Defence Research Agency
mikko.kiviharju@mil.fi

***Abstract*—Measuring the security of cryptographic systems (algorithms, protocols, software and hardware implementations etc.) is a difficult task. There does not exist one simple and easy to measure value that could be used to evaluate the relative strength of different cryptographic systems. On the other hand, there are more and more use cases where protections granted by cryptographic systems are needed. In some cases, there are certification and classification requirements for the use of cryptosystems that would benefit from good measures. Also new standards are being created for cryptography, usually based on competitions, where the proposals are evaluated based on some criteria. In this paper, we describe a taxonomy of the multiple metrics that can be associated with cryptographic systems and evaluate them based on a number of different attributes. We also reflect our taxonomy to the decisions made in several cryptographic standardisation competitions.**

Keywords–Cryptography; Security metrics; Taxonomy; Competitions; Cryptographic systems

I. Introduction

Cryptography is a key element in establishing trust in our digital society. Having reliable and correctly functioning cryptographic systems is necessary to realise many of the services that we all use in our everyday lives. Cryptography has thus become a crucial part of our critical infrastructures.

Cryptographic systems are built from different types of building blocks and designed to provide many different security goals depending on their anticipated usage. The security of the system depends on the theoretical algorithms and assumptions on their security proofs, the programming languages used to realise them, the platforms and operating systems that these programs utilise, and the hardware that runs all these. Thus, it is very difficult to give commensurate, yet simple measurements on the security of cryptographic systems.

Having such a simple metric would have great implications for developers and decision makers. A simple metric would benefit both standardisation and certification efforts that involve cryptographic systems and implementations. If an absolute metric could be devised, comparing different options would become a small exercise in comparing the values that these metrics give for different choices of cryptosystems. Alas, such a metric is not yet available and it might be nearly impossible to provide one.

However, there are many measures that are used to evaluate cryptographic protocols. The most notable one is the *key length* of a given algorithm. There are many reports, which give recommendations for key lengths for different algorithms in different contexts (e.g., [1]–[3]). These are mainly to be seen as lower limits for the key lengths of different cryptosystems and as such they offer only limited information on the security of a cryptosystem implementation. Existing efforts towards more comprehensive understanding of the traits of the cryptosystems by classifying cryptosystems from the metric perspective include metrics for algorithm security [4], and metrics from the attackers' point of view [5]. But a comprehensive metric, with commensurate components, is still not available.

In this paper, we survey the many different metrics for measuring the security of cryptographic systems and categorise them into four different categories. In Section II, we define some concepts used throughout this paper. In Section III, we discuss the properties of each measure and present an overview of our findings. For some measures, it is possible to have an ordering and for others it is not. We also study some competitions on cryptographic standards and how they have used different metrics in the decision making process in Section IV. Furthermore, in Section V, we discuss the possibilities, gaps and the necessity of having good metrics for cryptographic systems. Finally, in Section VI, we discuss the future work needed to realise better metrics for cryptographic systems and give conclusions of our research.

II. Attributes of Cryptographic Metrics

In this paper, we use the term *cryptosystem* to mean any algorithm or implementation that aims to provide cryptographic security for some defined target. Here, the cryptosystem can be a primitive, such as a hash function, or a fully-fledged file encryption software or a protocol for network security, e.g., Transport Layer Security (TLS) or something in between.

A metric is a way to measure some part or the totality of the security of a cryptosystem. A metric can have numerical values or it can be a qualitative description.

We also define some attributes that each metric can have. A metric is *measurable* if there is a standard convention on how the metric is measured and this is uniform across all applications of the metric (e.g., kilograms for weight). A metric is *semi-measurable* if there are several different conventions on how to measure the metric and some of these are not readily comparable with each other. In some cases the metric is *non-measurable*, which means that a standard for measurement does not exist or that the different values that the metric can have are not comparable in meaningful ways.

Another attribute is *practical relevance*. This measures how much the metric has practical relevance in evaluating the security of the cryptosystem. Some metrics are relevant in the theoretical frameworks and some metrics are more relevant in the practical world, where the cryptosystems are applied. An example of a very practical metric is the amount of memory required to attack a cryptosystem. A more theoretical metric is the proof framework, where a system is proven secure. In the theoretical world there is a big distinction, whether a proof is for example in the random oracle model or in the standard model, but the differences between these two do not manifest themselves as practical attacks in implementations.

We also make a distinction between *quantitative* and *qualitative* metrics. Quantitative metrics give a numerical or several numerical values to the cryptosystem and qualitative metrics give a description of the state of the cryptosystem.

III. Categories of cryptographic measures

This section presents our taxonomy of metrics for cryptographic systems. An overview of the taxonomy is presented in Table I.

A. Adversarial Model Metrics

Cryptology and especially cryptographic theory aims to formalize, how cryptographic algorithms work and withstand cryptanalysis. Due to the need for rigorous formalisms in cryptographic theory, the models used need to be very detailed, and yet general with respect to adversarial behaviour. We use the term *algorithmic metrics* to refer to metrics that involve cryptosystems independently of their realization in code or hardware. Algorithmic metrics are here divided to adversarial model metrics (Section III-A) and proof framework metrics (Section III-B).

As an example, consider the combination of the metrics in the following common concept: INDistinguishability under Chosen Ciphertext Attack or IND-CCA [6]. We observe here the following independent metrics:

- *Adversarial goal*: distinguish between random strings and actual ciphertext.
- *Adversarially available information*: a polynomial amount of information, before and after the cryptographic transformation.
- *Adversarial degrees of freedom of actions* include choosing the ciphertext-plaintext pairs adaptively (excluding the keys).

In addition to the three metrics above, we consider *adversarial resources*, which the designer of the cryptosystem expects the attackers to be able to wield. The four above metrics together are related to the *adversarial model*.

The adversarial resources consist of *computing power* and available *memory*. They are mostly well-defined and accessible metrics, with practical relevance.

Computing power is addressed here in both of its forms: exact attack complexities, and approximate or asymptotic complexities. Cryptographic theory rarely elaborates the adversarial models down to the detail of exact number of operations required to break the system. Instead, asymptotic estimates are given, and often even they are described only on the level of computational complexity classes.

In the case of exact complexities, values can be given, e.g., as the amount of floating point operations per second (FLOPS). In quantum computing, the unit can be based on, e.g., the amount of universal qubits and gates in the quantum computer. This metric is measurable and quantitative.

In the latter case, where the complexity class border is crossed, literature usually refers to different "computational models", the most common being Bounded-error Probabilistic Polynomial time (BPP), where polynomially bound, probabilistic Turing machines are expected. Other notable models include Bounded-error Quantum Polynomial-time (BQP) for quantum computers; and statistical, or unconditional security model, where the adversary is given limitless computational power. This metric is semi-measurable (as the exact relations between complexity classes are not known) and qualitative.

As the exact running time estimates can only be fixed once a cryptosystem is fully instantiated and parametrized, we consider this measure to consist of two subclasses of the whole: instantiated and non-instantiated computing power (asymptotic notations can be computed to exact metrics once the parameters, such as key size, are fixed).

Memory is the amount of memory that the attack requires. Analogously to the computing power, we divide this into two subclasses: instantiated (measurable and quantitative) and non-instantiated (semi-measurable and qualitative). Memory can also have some effect on the computing power needed for the attack. Some example memory complexity classes could be LOGSPACE and PSPACE.

Adversarially available information is the amount and type of data that the attack needs or is allowed for the adversary. We distinguish here at least six different types: *pre-crypto* (data before encryption, signing or other cryptographic transformation), *post-crypto, secret key-material* (symmetric or private asymmetric), *protocol runs, setup parameters* and *simulation environment master*. The four first ones are measured in bits, bytes or messages/keys/runs, the last two are discussed as follows:

- The access to *setup parameters* becomes relevant in cryptographic protocols, giving rise to, e.g., variants of Universal Composability (UC): Joint UC [7] and Global UC [8]. Possible value space could be {`local/global, per protocol/several runs`}.
- The control of the *master process*, which in cryptographic protocol security proofs models to what degree the adversary is able to control the (unspecified) protocol environment (resulting in yet other UC variants [9]). Possible value space could be {`Sim+Adv, Advonly, Env, *`}.

Adversarial goals need to be rigorously formalized, which usually results in case-specific definitions, and almost all values for the metric are incomparable, making it both qualitative and semi-measurable only. An example value space for typical goals is {`Semantic deduction, Information Leak, Local deduction, Global deduction, Total Break`}.

Adversarial degrees of freedom of action refer here to what the adversarial model is expecting the adversary to do. We

TABLE I. CATEGORIES AND PROPERTIES OF DIFFERENT METRICS OF CRYPTOGRAPHIC SYSTEMS.

Main category	Subcategory	Type	Measurable	Quantitative / Qualitative	Relevance
Adversarial model	Degrees of freedom		yes	Qualitative	P
		Corruption power Num. of principals	yes	Quantitative	T
		Corruption power Degree of corruption	semi	Qualitative	T
		Security game compliance	semi	Qualitative	T
	Adversarial available information	Pre-crypto	yes	Quantitative	P
		Post-crypto	yes	Quantitative	P
		Secret key material	yes	Quantitative	P
		Protocol runs	semi	Quantitative	P
		Setup parameters	semi	Qualitative	T
		Simulation environment	semi	Qualitative	T
	Adversarial goal		semi	Qualitative	P
	Adversarial resources	Computation power Instantiated	yes	Quantitative	P
		Computation power Non-instantiated	semi	Qualitative	T
		Memory Instantiated	yes	Quantitative	P
		Memory Non-instantiated	semi	Qualitative	T
Proof framework	Complexity & security assumptions		semi	Qualitative	T
	Abstraction assumptions	Type	semi	Qualitative	T
		Num. of assumptions	yes	Quantitative	T
		Maturity of assumptions	no	Qualitative	T
	Methodology	Tightness	yes	Quantitative	P
	Rigor		semi	Qualitative	T
Verification and maturity	Key length	Bits for criteria compliance	yes	Quantitative	P
	Assurance levels	Assurance standard or profile	yes	Qualitative	P
		Level, e.g. EAL	yes	Quantitative	P
	Coverage	Percentage of tests	yes	Quantitative	P
	Method efficiency	Number of detected vulnerabilities	yes	Quantitative	P
	Human efficiency	Academic research	semi	Quantitative	P
	Verification time	Time since released for evaluation	yes	Quantitative	P
		Size and efforts of eval. community	yes	Quantitative	P
	Readiness level	Technology readiness level	yes	Quantitative	T
		Integration readiness level	yes	Quantitative	T
		System readiness level	yes	Quantitative	T
		PETS maturity model	yes	Qualitative	P
Cost and performance	Time costs	Execution overhead	yes	Quantitative	P
		Communication overhead	yes	Quantitative	P
	Memory costs	Run-time	yes	Quantitative	P
		Storage	yes	Quantitative	P
		Communication	yes	Quantitative	P
	Implementation complexity	Size of software	semi	Qualitative	P
		Dedicated hardware requirements	semi	Qualitative	P
	Energy efficiency	Algorithm complexity dependent	yes	Quantitative	P
		Hardware platform dependent	yes	Quantitative	P

propose to divide the degrees of freedom into three: *General, Corruption power* and *Game compliance*.

Corruption power. In interactive protocols, the adversary is also assumed to be able to access and/or modify the private information of some of the principals. This is called corruption, and depending on the scheme, only a certain number of principals are allowed to be corrupted. Sometimes even more fine-grained "corruptive power" is allowed [10]. The example values of this metric could include a (quantitative) percentage of corrupted principals and a (qualitative) description of the degree of corruption within one principal (see [10] for subprotocol-level detail).

For the *general metrics*, cryptographic formalisms differ in the amount of principals: Single-party settings (conventional encryption and signatures) and multiparty settings (protocols). As we show below, the multi-party setting does not bring that many new metrics per sé.

In the single-party setting, only one or fixed, integral set of cryptographic transformations (a black box) are usually considered. In this case, the adversary may be able to *observe* some or all of the inputs, or to *choose* (possibly adaptively) some or all of them. Note that we consider modification of inputs and other adversarially available information to belong to the "choosing" process. Some of the possible values in the single-party setting would then be `'Observe'`, `'Choose'` and `'Choose adaptively'`, in increasing order.

In the multi-party setting, i.e., protocols, the situation with adversarial behaviour appears at first sight to be more complex, as the security models are more varied. In the Dolev-Yao model [11], the principle is that the "attacker carries the message", or that the adversary is free to read, modify, add and delete protocol messages and corrupt protocol principals (in effect stealing their private key material). However, the convention we made in the single-party setting already covers the deletion, modification and adding of protocol messages, global setup parameters modification and protocol environment control, since the ability to choose message (/parameters/environment properties) for a single party translates to all of the above. We thus conclude that we have not identified more metrics from the multi-party setting.

Game compliance. Many of the formalisms in cryptographic security can be divided into two: game-based approaches and simulation-based approaches. Game-based approaches are basically a protocol, which try to model the adversarial behaviour in some commonly thought scenarios. Simulation-based approaches try to enable showing security irrespective of the adversarial behaviour. The best the attacker can do, is to perform the idealized, non-cryptographic tasks assigned to replace crypto in the simulation (SIM) model.

In the metrics, we consider this distinction to be an adversarial degree of freedom in the sense that the adversary is either constrained to follow some security game protocol, or not. The values could be, for example {`Game-App`, `Game-Gen`, `SIM`}, making a further distinction between general security games and very application-specific games.

B. Security Proof Framework Metrics

Proof framework is the framework in which the security proof is conducted. This includes multiple assumptions (for abstractions of certain functions and for the complexity of several mathematical problems), the rigor used and the proof methodology.

A metric clearly tied to the proof methodology is *tightness* of the proof. This concept indicates, how exactly the resource needs for different phases of the proof are estimated. This metric is measurable and quantitative, as typical asymptotical $O(f(n))$ expressions are used here.

Complexity assumptions are the foundation of many types of cryptographic proofs. They are assumptions on the hardness of different mathematical problems, usually that their time-complexity is superpolynomial in the security parameter. These assumptions can have several metrics:

- Assumption's time-complexity. The metric is measurable, quantitative and practical, as it directly affects key size. The metric is expressed with the Big-Oh-notation (e.g., $O(f(n))$).
- Type, if the assumption belongs to a known sequence of implications (e.g., Decisional Diffie-Hellman (DDH) $\Leftarrow$ Computational Diffie-Hellman (CDH) $\Leftarrow$ Discrete Log (DL) problem.) A possible common labelling borrows from the general ordering for several problems, where decisional problems (DDH) are usually easier than computational problems (CDH), and finally the primitive inversion problem (DL): {`decisional`, `computational/search`, `inversion`}. This metric is semi-measurable and qualitative.

Abstraction assumptions cover, how much the proof methodology uses abstractions, what kind of type they present and their maturity. Typical abstractions give different functions as ideal oracles, the most famous probably being the Random Oracle Model (ROM, [12] with variations in [13] and [14]). Many other oracles exist as well, e.g., the Generic Group Model (GGM) [15], the Ideal Cipher Model (ICM) [16], and the Common reference string model [17]. Sometimes the oracles are implicit, such as the Dolev-Yao modelling on encryption operations (which are assumed to be secure). If no abstractions are used, the proof is said to be conducted in the Standard Model (SM).

The actual metrics are proposed as follows:

- The number of abstractions used. For a proof in SM this would be zero. Different abstractions would be weighed differently depending on their maturity and suitability for the cryptosystem
- Assumption maturity (we consider this to be in the verification category and not elaborated more here)
- Type. Not all of the abstraction are equal, as there are some known relations among them (e.g., ICM and ROM have been proven equal in some cases [18]). We then postulate, that like with the complexity assumptions, there is a common metric able to classify abstraction assumptions as well, but we leave it for future study.

Rigor refers to the level of detail of the proof, its compliance to commonly used proof techniques and the assurance in the validity of the proof. Many schemes outside the cryptologic community often rely on pure heuristics, others merely state that the scheme is essentially similar to an earlier scheme and overlook the security proof completely. Many other systems are too complex to contain fully rigorous proofs in single conference papers, making the authors only outline the proofs. Ideally, proofs should be fully detailed, and externally verified. The value space for this metric would then be {`Heuristic`, `Referenced`, `Outlined`, `Full`, `Verified`}.

C. Verification and maturity metrics

The strength and correctness of cryptographic implementations can be verified with different testing methods and tools. For instance, independent or national laboratories have product certification frameworks and programs for verifying that implementations have required functionality and behave as expected with different inputs.

As already mentioned in the introduction, *key length* is one of the most used metrics for cryptosystem security. In our taxonomy, key length considers the maturity and verification level that a cryptosystem has. It is an indicator that shows if the security parameters of a cryptosystem are up to the standards, which are defined for that cryptosystem and its use. It is a measurable, quantitative and practical metric.

Assurance Levels are measurements indicating system's security when compared against common or standard evaluation and testing requirements. For instance, Evaluation Assurance Level (EAL) is a seven point-scale metric used by the Common Criteria (CC) [19] security evaluation framework for implementations; Common Criteria's Protection Profile is simpler two point-scale (compliant/non-compliant) metric for specific product categories; Cryptographic Algorithm Validation Program (CAVP) [20] defines functional and statistical tests for algorithms with a two-point (pass-fail) scale; Cryptographic Module Validation Program (CMVP) [21] defines validation tests for hardware implementations in four point scale (i.e., FIPS 140-2 security levels); and ISO 29128 [22] Protocol Assurance Levels define requirements for the scope and automation of formal modelling and verification of cryptographic protocols. In addition to the generic frameworks, there also exist frameworks that are specific for industry field or for an area of cryptography. For instance, the Payment Card Industry [23] has defined its own test requirements for two point scale evaluation of cryptographic hardware modules and National Institute of Standards and Technology (NIST) has specified [24] a large suite for randomness testing. Verification metrics describe the coverage and effectiveness of the verification and testing actions that the cryptographic product has passed. Assurance levels are semi-measurable and quantitative metrics.

Coverage refers to the percentage of potentially vulnerable areas that are tested or verified. Coverage is complete coverage if every area with potential vulnerabilities are verified. The areas that can be tested include, e.g., functionality, interfaces and protocols, randomness, susceptibility to side-channel and fault injection attacks, life-cycle, as well as susceptibility to physical tampering and to reversing of obfuscated functionality attacks. Existing test suites, validation program requirements or common criteria profiles can be utilized when estimating whether all relevant areas are included to verification and whether all tests for the relevant areas are executed. This metric is measurable and quantitative.

Effectiveness of verification methodologies - such as requirement specifications, test patterns, statistical testing tools, formal analysis methods, and simulation tools - refers to design or implementation failures that can be detected with the given methodology. The effectiveness depends on the available software and hardware facilities, as well as on the quality of the processes in the evaluating community or laboratory. A straightforward quantitative and measurable metric of effectiveness is the amount of failures that are detected with the method. When different testing methods are available, it is also possible to estimate false positive and false negative ratios.

Effectiveness of human verification depends on the skills of human evaluators for verification and testing. These capabilities can be measured, e.g., by looking at the experience and education of evaluators, as well as past performance and reputation. Quantitative and measurable metrics for human verification include experience in years, number of performed evaluations, as well as the scientific author metrics (number of fresh related publications).

Verification time refers to the hours, months, or years that have been spend on exploring the cryptographic solution against vulnerabilities. Time accumulates from intensive product evaluations as well as from the verification and testing by scientific and user community during the system lifetime. The older and more dispersed the system is, the less unknown weaknesses it is likely to have. This is a quantitative and measurable metric.

The maturity metrics measure how ready and suitable a cryptosystem is. This can be a metric for a specific component as in Technology Readiness Levels (TRL) or a more comprehensive metric of a whole systems, such as the PETS maturity metric [25]. Some of these metrics are more complex derivations of the TRL, such as Systems Readiness Level (SRL) [26] and Integration Readiness Level (IRL) [27].

TRL measures the readiness of a single component. This metric is measurable and quantitative. *IRL* measures the readiness of components to be integrated to form a more complex system. This metric is measurable and quantitative. *SRL* measures the readiness of a complete system based on the TRLs and IRLs of the different components. The metric is measurable (if normalized) and quantitative.

PETS maturity metric is a measure for the quality and readiness of privacy enhancing technologies [25]. The measurement is carried out with both measurable indicators (such as the number of papers/patents and lines of code) and a more heuristic evaluation by experts. There is a defined procedure on how to reach consensus on possibly differing evaluations by experts. In some sense, this is similar to the jury evaluation used in some cryptographic standardisation competitions. In the PETS maturity metric, the evaluation is open and transparent, whereas in some cryptographic competitions this is not the case. The PETS maturity metric is measurable and qualitative.

D. Cost and performance metrics

The feasibility of cryptographic products depends not only of their security strength, but also other factors that are measured using cost and performance metrics. Cost and performance metrics can be calculated for the whole system or separately for an individual role (e.g., decrypter, encrypter, signer, verifier). Asymmetric cost division between roles may be beneficial, e.g., in cloud or Internet of Things scenarios where another party has more resources available for cryptographic operations.

Time costs originate from the computations, such as key generation, encryption and decryption, as well as public and private key operations, and from communications, where cryptography causes additional overhead, expands communication and negotiations. Time costs can be estimated by counting elementary operations that a cryptographic solution implies. This is a quantitative and measurable metric.

Size costs relate to the need for run-time and storage memory, as well as to the communication bandwidth. They depend on the sizes of keying material, ciphertexts, and signatures, as well as on run-time memory requirements of algorithms. This is a quantitative and measurable metric.

Implementation complexity relates to the size and costs of software or hardware implementations. A key attribute is whether the solution is suitable for standard computing platforms (e.g., Intel x86 or ARM-based) or whether it requires specialized hardware. An important attribute is also whether the performance of the algorithm can be improved with special hardware, such as parallel platforms or extended instruction sets. Complexity can be estimated either by counting lines of code or by counting required hardware resources like gate counts. This is a semi-measurable (as there are many ways to measure complexity) and qualitative metric. *Energy efficiency* depends on use of computing, memory, and communication resources and their cost in different platforms. This is a measurable and quantitative metric.

IV. ANALYZING METRICS IN CRYPTOGRAPHIC COMPETITIONS

One way to evaluate our taxonomy of metrics is to take a look at the different competitions for cryptosystems. For this, we evaluated 8 different competitions and the rationales that they used to select the winning algorithm(s). The chosen competitions are Advanced Encryption Standard (AES) [28], New European Schemes for Signatures, Integrity, and Encryption (NESSIE) [29], Cryptography Research and Evaluation Committees (Cryptrec) [30], the ECRYPT Stream Cipher Project (eStream), NIST hash function competition [31], Password Hashing Competition (PHC) [32], Competition for Authenticated Encryption: Security, Applicability, and Robustness (CAESAR) [33], and Post-Quantum Cryptography Standardization (PQCS) [34].

A. AES competition

The goal of NIST's AES competition was to specify an unclassified, publicly disclosed encryption algorithm capable of protecting sensitive government information well into the next century. The algorithm would have to implement symmetric key cryptography as a block cipher and support a block size of 128 bits and key sizes of 128, 192, and 256 bits.

Competition started in January 1997 and lasted 46 months. There were 21 submissions. The winner was Rijndael which was renamed to AES, the other finalists were Serpent, Twofish, RC6 and MARS. Initial evaluation criteria were security, cost, and algorithm and implementation characteristics. Final criteria were general security, software implementations, restricted-space environments, hardware implementations, attacks on implementations, encryption versus decryption, key agility, other versatility and flexibility, and potential for instruction level parallelism [35].

NIST stated that the most emphasis in AES competition was on security. Rather than having government agencies scientists test and measure the security of each algorithm, they asked the public cryptographic community for help. Members of the cryptography community tested each algorithm's resistance to cryptoanalysis. Cryptoanalysis included testing each algorithm's security against known practical and theoretical attacks. The public also analysed the algorithms by determining the mathematical soundness. None of the finalists were statistically distinguishable from a random function. The team at NIST carefully considered the public analyses and used the results of these analyses when evaluating the algorithms. Measuring cost included licensing requirements, computational efficiency, and memory requirements. The algorithm characteristics and implementation criteria included flexibility, hardware and software suitability, and simplicity [36].

There were no known attacks on any of the five finalists at the time of the judging, so other, less palpable measures were used to determine security of the ciphers. When comparing hardware and software performance of the finalists, Rijndael and Twofish exhibited very similar results. The Rijndael implementation was clearly the simpler of two and shared close ties with an ancestor cipher, Square, that had received a significant amount of analysis, while Twofish had no such ancestor [28].

B. Cryptography standardization projects

NESSIE and Cryptrec projects were inspired by the AES competition. The goal of the NESSIE project was to identify secure cryptographic primitives. Competition started in March 2000 and lasted 36 months. There were 42 submissions and twelve algorithms were chosen for the final portfolio. Initial evaluation criteria were long-term security, market requirements, efficiency and flexibility.

The goal of the Cryptrec project was to evaluate and recommend cryptographic techniques for government and industrial use. Competition started in May 2000 and lasted 34 months. Out of 63 submissions, ten algorithms were chosen for the final portfolio. Initial criteria for evaluation were security, cost, and algorithm and implementation characteristics.

The goal of eStream project was to identify new stream ciphers suitable for widespread adoption, because in the NESSIE project all stream ciphers failed. Competition started in October 2004 and lasted 43 months. There were 34 submissions and seven algorithms were chosen in the final portfolio. Initial criteria were security, performance, simplicity and flexibility, justification and analysis, and quality of documentation.

C. Hash algorithm competitions

The goal of the NIST hash function competition was to develop a new hash function called Secure Hash Algorithm 3 (SHA-3). The competition started in November 2007 and lasted 60 months. There were 51 submissions. The winner was Keccak which was renamed to SHA-3. Other finalists were BLAKE, Grøstl, JH, and Skei. Initial criterion were security, cost, and algorithm and implementation characteristics. Final criteria were performance, security, analysis and diversity [31].

The goal of PHC was to find password hash functions that can be recognized as a recommended standard. Competition started in January 2013 and lasted 31 months. There were 24 submissions. The winner was Argon2 and the other finalists were Catena, Lyra2, Makwa, yescrypt. Initial criterion were security, efficiency, and simplicity. Final criteria were defence against GPU/FPGA/ASIC attacks, defence against time-memory tradeoffs, defence against side-channel leaks, defence against cryptanalytic attacks, elegance and simplicity of design, quality of the documentation, quality of the reference implementation, general soundness and simplicity, originality and innovation [32].

D. Recent competitions

The goal of CAESAR competition was to find new authenticated ciphers in three different categories. The competition started in January 2013 and lasted 74 months. There were 57 submissions and six were chosen for the final portfolio. The winners in the three use cases were Ascon for lightweight applications, AEGIS-128 for high-performance applications and Deoxys-II for defence in depth [33].

The goal of PQCS competition [34] is to standardize post-quantum cryptography to replace the current public key cryptosystems that can be broken with a quantum computer, e.g., [37]. The competition started in January 2017 and got 69 submissions. Initial criteria were security, cost and performance, and algorithm and implementation. The goal is to find suitable methods for digital signatures and key exchange, which are the two major use cases for public key cryptography.

E. Comparative analysis

Many of the security metrics, which were presented in the previous section, were present in the standardization competitions in different ways. As there has not been a uniform approach over the different competitions towards the metrics that our taxonomy describes, we have condensed the view especially with regards to the adversarial model and proof framework categories.

Table II summarizes our interpretation of the mapping between metrics and competitions. We looked at the metrics from three perspectives. First, we considered the main motivations to (new) standards, i.e., whether a low value for the metric in a predecessor standard (or a missing standard) was the reason for starting the competition in the first place. There were two main motivations: the development of adversarial capabilities which obsoleted earlier standards and the new security goals previously unaddressed by standards. Motivations are marked in the table with M. Secondly, we looked if competitions explicitly or implicitly expressed qualitative or quantitative requirements as their selection criteria. The existence of these metrics are marked in the table with E and I, respectively. Thirdly, we looked at publicly available measurable statistics that illustrated the verification efficiency of competitions. In particular, we evaluated the effectiveness of human verification by looking at the number of scientific articles that were published during the competition or the year after and that were returned by a Google Scholar query: "'competition name' candidate cryptography", which was performed on May 10th 2019. The relation between the amount of winners and candidates was listed as a metric of verification method; it is not an indication of failure detection rates but an indication of interest, which often leads to better results.

V. Discussion

Measuring the security of systems is still a very difficult task even though the area has been researched for a long time and the interest has been increasing in recent years. Measuring the security and strength of cryptosystems seems to be even harder, because there are so many different metrics and variables involved and these also interact in many ways. Furthermore, the notion of security is very much context dependent. Even a secure cryptographic primitive used in a wrong context provides very little security. An insecure primitive in a wrong context provides even less security, e.g., [38].

Cryptographic metrics have lots of interdependencies. Some metrics directly or indirectly derive from other metrics, while other metrics are atomic responses to one particular requirement, capability, or threat. For instance, the key length is a derived metric whose value is motivated by the development of adversarial resources, limited by cost metrics, and defined by security proofs, assumptions and goals. In general, there is a trade-off between cost and performance metrics and many of the algorithmic metrics. Time and memory costs decrease directly with the corresponding adversarial resources. Higher adversarial capacities necessitates higher key lengths, complexity, stronger proofs and computation models. Each maturity metric depends directly on several of the algorithmic and verification metrics.

To illustrate the complexity of the situation, we have generated Figure 1. It depicts the different dependencies we have found in our taxonomy as arcs. The arcs are asymmetric and should be read left-to-right on the top part of the figure and right-to-left on the bottom of the figure. Bolder arcs indicate a more linear relationship, while thinner and lighter arcs indicate

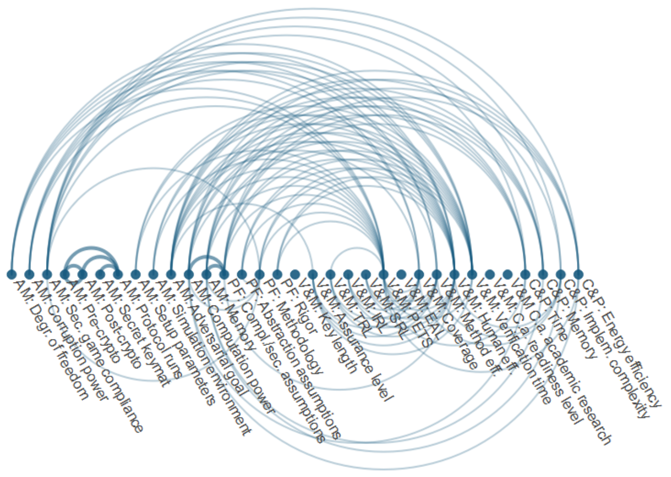

Figure 1. The complexity of the inter-metric dependencies.

that the dependency is not very linear. The abbreviations stand for Adversarial Model (AM), Proof Framework (PF), Verification and Maturity (V&M) and Cost and Performance (C&P).

Our taxonomy is open for new inter-metric derivatives. For instance, there are some measures that we have chosen, due to simplicity, not to present in our taxonomy as their own separate metrics. One is the total (monetary) cost of an attack as a resource metric. It could be argued that this is one of the most relevant metrics there is. On the other hand, it is also derivative of the attack resource metrics that have been included in our taxonomy. Another such quantity is time, which we see as a part of the computing power metrics.

Competitions must grade each candidate in order to determine the winner. This overall grade must be one-dimensional and cannot be formulated directly by counting averages, as units of measurements are not uniform and different competitions have had different valuations for their metrics. Formal means for grading exist. In particular, weighted average sum of each measurement is multiplied with a weight factor, which normalizes the scale and reflects metric's importance, and then divided with the sum of weight factors. The difficulty is in deciding which metrics get the most weight and how the qualitative metrics incorporate into the whole measurement. The decision regarding the weights of different metrics is contextual and depends on threat models and use cases. Qualitative metrics transform into numerical values with level-based grading criteria, which also depends on the context. In addition, thresholds are used in overall grading: some metrics must reach some minimal level to enable eligibility but do not affect overall grade.

One limitation of the standardisation competitions is that they usually consider only quite low level cryptosystems (block/stream ciphers, hash functions etc.) and not more complex systems or protocols. More complex cryptosystems, such as the TLS protocol, are formed through more traditional standardisation efforts. There we might expect to see different utilisation of the different metrics and perhaps more complex rationale for the decisions that are made. It seems from our results, that much of the consideration is given also to the cost

TABLE II. METRICS IN CRYPTOGRAPHIC STANDARDIZATION COMPETITIONS

Metric	AES	NESSIE	CRYPTREC	eStream	SHA-3	PHC	CAESAR	PQCS
Adversarial model	M	-	-	-	M	-	-	M
Key length	128,192,256	-	-	128, 80	-	-	-	-
Complexity	I (security analysis by community)							
Security goal	E	M,E	M,E	M,E	E	M,E	E	E
Computational model	Classical							and Quantum
Method efficiency (winners/candidates)	1/21	12/42	10/63	7/34	1/51	1/24	6/57	?/69
Human efficiency (peer-review papers)	539	179	349	46	1400	83	1400	803
Verification time	1997-2001	2000-03	2000-03	2004-09	2007-13	2013-17	2014-19	2016-
Readiness level	TRL 4 (prototype for laboratory validation)							
Time costs	E	E	E	M,E	E	E	E	E
Memory costs	E	I	E	M,E	E	I	E	E
Implementation complexity	E	E	E	E	E	E	M,E	E
Energy efficiency	I	I	I	I	I	I	I	I

and performance metrics. This is understandable because the winning algorithms are to become widely deployed standards.

In practice, none of the competitions has explicitly formulated their selection criteria or weights. Instead, all competitions have provided high-level instructions and delegated final grading to individual jurors who may have their own criteria. Typically, competitions explicitly or implicitly specify security goals, as well as adversarial models and assumed resources, but requirements related to other security and feasibility metrics are qualitative with few exceptions such as the key length targets for eStream and AES.

VI. CONCLUSION AND FUTURE WORK

This paper describes a taxonomy of different types of metrics that can be used to evaluate the security of cryptosystems. Our taxonomy has four categories: Adversarial model metrics, Proof framework metrics, Verification and maturity metrics, and Cost and performance metrics. It can be seen that different metrics have different attributes and that many of the most relevant metrics are not easy to measure and compare.

The evaluation of the competitions for cryptographic standards against our taxonomy shows that there are many commonalities between competitions, but there is no predefined set of metrics that the submissions are evaluated against. Thus, there is a need to continue research in this direction. This should lead towards a more standardised set of relevant and easy to use metrics for cryptosystems.

There is a lot of room for future work in this area. Our taxonomy provides an overview of the different metrics and their attributes, but we are still a long way from building and proposing a comprehensive metric for measuring cryptosystems. This is the major goal for future work in this area. One possible interesting direction for future research could be to utilise the TRL, IRL and SRL measures to more complex cryptosystems. The concepts have been tested in other fields, but in cryptography and cryptosystems they have not been studied yet. If this approach would be successful, we would have a potentially very general metric for measuring cryptosystems that could be used in many different contexts.

VII. ACKNOWLEDGEMENTS

This research has been funded by Defence Forces Research Program 2017 (PVTO 2017).

REFERENCES

[1] N. P. Smart, V. Rijmen, B. Gierlichs, K. Paterson, M. Stam, B. Warinschi, and G. Watson, “Algorithms, key size and parameters report,” European Union Agency for Network and Information Security, 2014, pp. 0–95.

[2] BSI, “Cryptographic mechanisms: Recommendations and key lengths,” https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.pdf, BSI TR-02102-1, Tech. Rep., 2018.

[3] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid, “NIST special publication 800-57,” NIST Special Publication, vol. 800, no. 57, 2007, pp. 1–142.

[4] N. Jorstad and T. S. Landgrave, “Cryptographic algorithm metrics,” 20th National Information Systems Security, 1997. [Online]. Available: http://csrc.nist.gov/nissc/1997/proceedings/128.pdf

[5] Z. Benenson, U. Kühn, and S. Lucks, “Cryptographic Attack Metrics,” in Dependability Metrics. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 133–156.

[6] M. Bellare and P. Rogaway, “Optimal asymmetric encryption,” in Advances in Cryptology—EUROCRYPT’94. Springer, 1995, pp. 92–111.

[7] R. Canetti and T. Rabin, “Universal composition with joint state,” in Annual International Cryptology Conference. Springer, 2003, pp. 265–281.

[8] R. Canetti, Y. Dodis, R. Pass, and S. Walfish, “Universally composable security with global setup,” in Theory of Cryptography Conference. Springer, 2007, pp. 61–85.

[9] A. Datta, R. Küsters, J. C. Mitchell, and A. Ramanathan, “On the relationships between notions of simulation-based security,” in Theory of Cryptography Conference. Springer, 2005, pp. 476–494.

[10] R. Canetti, “Universally composable security: A new paradigm for cryptographic protocols,” in Proceedings 2001 IEEE International Conference on Cluster Computing. IEEE, 2001, pp. 136–145.

[11] D. Dolev and A. Yao, “On the security of public key protocols,” IEEE Transactions on information theory, vol. 29, no. 2, 1983, pp. 198–208.

[12] M. Bellare and P. Rogaway, “Random oracles are practical: A paradigm for designing efficient protocols,” in Proceedings of the 1st ACM conference on Computer and communications security. ACM, 1993, pp. 62–73.

[13] P. Ananth and R. Bhaskar, “Non observability in the random oracle model,” in International Conference on Provable Security. Springer, 2013, pp. 86–103.

[14] J. B. Nielsen, “Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case,” in Annual International Cryptology Conference. Springer, 2002, pp. 111–126.

[15] V. Shoup, “Lower bounds for discrete logarithms and related problems,” in International Conference on the Theory and Applications of Cryptographic Techniques. Springer, 1997, pp. 256–266.

[16] C. E. Shannon, “Communication theory of secrecy systems,” Bell system technical journal, vol. 28, no. 4, 1949, pp. 656–715.

[17] R. Canetti and M. Fischlin, “Universally composable commitments,” in Annual International Cryptology Conference. Springer, 2001, pp. 19–40.

[18] J.-S. Coron, T. Holenstein, R. Künzler, J. Patarin, Y. Seurin, and S. Tessaro, “How to build an ideal cipher: the indifferentiability of the feistel construction,” Journal of cryptology, vol. 29, no. 1, 2016, pp. 61–114.

[19] Common Criteria, “Common Criteria for Information Technology Security Evaluation Part 2 : Security functional components,” Security, no. April, 2017, pp. 1–323.

[20] NIST, “Cryptographic Algorithm Validation Program,” 2018. [Online]. Available: https://csrc.nist.gov/Projects/Cryptographic-Algorithm-Validation-Program

[21] NIST, “Derived Test Requirements for FIPS PUB 140-2, Security Requirements for Cryptographic Modules,” 2011.

[22] International Organization for Standarization, “ISO/IEC 29128:2011 - Information technology – Security techniques – Verification of cryptographic protocols,” 2011.

[23] PCI Security Standards Council, “Payment card industry data security standard v3.0,” 2013, https://www.pcisecuritystandards.org/security_standards/documents.php.

[24] NIST, “Special publication 800-22. a statistical test suite for random and pseudorandom number generators for cryptographic applications,” 2010.

[25] M. Hansen, J. Hoepman, M. Jensen, and S. Schiffner, “Readiness analysis for the adoption and evolution of privacy enhancing technologies: Methodology, pilot assessment, and continuity plan,” Tech. rep., ENISA, Tech. Rep., 2015.

[26] B. Sauser, D. Verma, J. Ramirez-Marquez, and R. Gove, “From trl to srl: The concept of systems readiness levels,” in Conference on Systems Engineering Research, Los Angeles, CA, 2006, pp. 1–10.

[27] B. Sauser, R. Gove, E. Forbes, and J. E. Ramirez-Marquez, “Integration maturity metrics: Development of an integration readiness level,” Information Knowledge Systems Management, vol. 9, no. 1, 2010, pp. 17–46.

[28] E. Chu, F. Liu, J. Yu, J. Sharma, P. Kim, and P. Kim, “Selection of advanced encryption standard,” MIT, Tech. Rep., 2000.

[29] B. Preneel, A. Biryukov, C. De Cannière, S. Örs, E. Oswald, B. van Rompay, L. Granboulan, E. Dottax, G. Martinet, S. Murphy et al., “New european schemes for signatures, integrity, and encryption,” European Project IST-1999-12324, 2004.

[30] Information Technology Promotion Agency, Telecommunications Advancement Organization of Japan, “CRYPTREC report 2002,” 2003.

[31] S.-j. Chang, R. Perlner, W. Burr, M. Turan, J. Kelsey, S. Paul, and L. Bassham, “Third-round report of the sha-3 cryptographic hash algorithm competition,” NIST, Tech. Rep., 2009.

[32] J. P. Aumasson, “Password hashing competition,” https://password-hashing.net/, April 2019.

[33] D. J. Bernstein, “Caesar submissions,” https://competitions.cr.yp.to/caesar-submissions.html, March 2019.

[34] G. Alagic et al., Status Report on the First Round of the NIST Post-Quantum Cryptography Standardization Process, NIST Internal Report 8240. US Department of Commerce, National Institute of Standards and Technology, 2019.

[35] D. J. Bernstein, “Aes: the advanced encryption standard,” https://competitions.cr.yp.to/aes.html, January 2014.

[36] J. Nechvatal, E. Barker, L. Bassham, W. Burr, M. Dworkin, J. Foti, and E. Roback, “Report on the development of the advanced encryption standard,” Journal of Research of the National Institute of Standards and Technology, 2001.

[37] P. W. Shor, “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer,” SIAM Review, vol. 41, no. 2, jan 1999, pp. 303–332.

[38] F. Y. Rashid, “Adobe’s hacked passwords: They are terrible!” https://uk.pcmag.com/opinion/12060/adobes-hacked-passwords-they-are-terrible, November 2013.