Computer Communications 52 (2014) 47–59

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Mean privacy: A metric for security of computer systems [☆]

Jaafar Almasizadeh, Mohammad Abdollahi Azgomi [*]

Trustworthy Computing Laboratory, School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran

ARTICLE INFO

Article history:
Received 13 March 2013
Received in revised form 13 April 2014
Accepted 27 June 2014
Available online 4 July 2014

Keywords:
Computer systems
Security
Privacy
Quantification
Information theory

ABSTRACT

In this paper, we propose a new approach for quantitative security analysis of computer systems. We intend to derive a metric of how much private information about a computer system can be disclosed to attackers. In fact, we want to introduce a methodology in order to be able to quantify our intuitive interpretation of how attackers act and how much they are predictable. This metric can be considered as an appropriate indicator for quantifying the security level of computer systems. We call the metric "Mean Privacy" and suggest a method for its quantification. It is quantified by using an information-theoretic model. For this purpose, we utilize a variant of attack tree that is able to systematically represent all feasible malicious attacks that are performed to violate the security of a system. The attack tree, as the underlying attack model, will be parameterized with some probability mass functions. The quantitative model will be used to express our intuition of the complexity of the attacks quantitatively. The usefulness of the proposed model lies in the context of security analysis. In fact, the analysis approach can be employed in some ways: Among several options for a system, we can indicate the most secure one using the metric as a comparative indicator. The security analysis of systems that operate under a variety of anticipated attack plans and different interaction environments can be carried out. Finally, new security policies, countermeasures and strategies can be applied to increase the security level of the systems.

1. Introduction

Ideally, we would like to have computer systems that are completely secure. Unfortunately, past and current experiences have shown that even the best attempts to build secure systems are not perfect. Generally speaking, the main sources of insecurity in computer systems are as follows. First, during the design and development phases, it would not be feasible to make a computer systems' software and hardware components completely free of vulnerabilities. These vulnerabilities would be suitably exploited to make the systems insecure. Second, the interaction environment of systems with users is very complex. Thus, the behavior of malicious attackers cannot be well understood. Third, with freely available attacking tools, conducting a large number of attacks against systems is a highly automated process. As a result, compared with the past situation, it requires a lower skill level and has a higher probability of success. Fourth, human-made errors leading to security holes and flaws frequently occur. Usually, such errors stem from the lack of sufficient experience and professional competence. These errors, due to their human nature, are always expected to be present, more or less.

In fact, the big advantage that attackers have is that they need to find only a single weakness of a system, while the system administrator must know and cover all weaknesses of the system to obtain perfect security. We need to analyze security from a relative point of view. As a practical matter, it is strongly accepted that an absolute system security level cannot be achieved. Therefore, it is recommended to consider security as a quality of service (QoS) attribute of systems (e.g., availability, reliability, delay, etc.) so as to be able to quantify how much their security level can be achieved. In other words, to validate the efforts made for securing systems, it is necessary to obtain a quantitative metric indicating their security level. Such a metric can characterize how good the security efforts are and how much they can be trusted.

Ideally, the process of quantitative security evaluation of a computer system can answer the following questions: With an initial design, is the system needed to be redesigned to meet its security requirements? With a default configuration, is the system needed to be reconfigured to meet the security requirements? Can the system provide a specified security level and meet the security requirements? By how much can the desirable security be achieved? What is the tolerance threshold of the system when

[☆] This research was supported by the Research Institute for ICT of Iran.

[*] Corresponding author. Address: School of Computer Engineering, Iran University of Science and Technology, Hengam St., Resalat Sq., Tehran 16846-13114, Iran. Fax: +98 21 73225322.

E-mail address: azgomi@iust.ac.ir (M. Abdollahi Azgomi).

http://dx.doi.org/10.1016/j.comcom.2014.06.009
0140-3664/© 2014 Elsevier B.V. All rights reserved.

confronted with attackers, that is, what the intensity level of malicious attacks can be handled and responded by the system security mechanisms? Answering such questions is a very useful task because it can provide more insights into security issues of a system under study. With the answers in hand, the security experts can make appropriate decisions of how to change existing security policies, services and mechanisms and define new ones.

To address such issues, we seek to find a way for the prediction of the complexity level of an attack process against a system. It is desirable to estimate how much attack scenarios are predictable. We define a new metric for the security analysis of computer systems that is appropriately consistent with our purposes. We propose the term "Mean Privacy" and a way for its quantitative evaluation. It is a predictive measure of the level of diversity and sophistication of attack scenarios. In fact, it is a good criterion that can be used to quantify the intuitive view of how attackers may behave. Clearly, in practice, we are not able to obtain exact values for security metrics. Indeed, in most situations, the computation of the exact values of the metrics is neither feasible nor necessary. The essential issue is to derive a reliable criterion so that it will be possible to quantitatively compare systems from the security standpoint.

For this purpose, we suggest a probabilistic model for describing an attack process and computing the respective metric. We employ some ideas from the field of information theory and apply them to our problem. The main idea behind this paper is to extend the concept of entropy to perform security analysis. In fact, this paper is our initial trends towards utilizing the fundamental concepts of information theory for developing a systematic method of security evaluation. In order to compute the quantitative metric, the proposed attack model needs to be parameterized using suitable data. The input parameters to the model are attack data and the output of the analysis process of the model is the desirable metric.

This paper is generally organized as follows. In Section 2, related works on security metrics are introduced. In Section 3, the details of the proposed modeling approach, step by step, are explained. In order to clarify the underlying concepts of the model and its applications, an illustrative example is given in Section 4. Finally, conclusions and future works are presented in Section 5.

2. Related work

The challenging area of quantitative security evaluation has been received much more attention in recent years. In this section, we want to provide a general overview of the most important methods worked out in this important area of security analysis. Due to the incomplete administrator's knowledge of the behavior, intent and skill level of attackers, the prediction of their behavior would be a very difficult task. That is, due to the uncertainty in the attack process, it cannot be completely captured. Thus, the use of probabilistic and stochastic models and concepts in the area of security evaluation can be an appropriate approach. As will be shown, stochastic and probabilistic modeling techniques used in the context of dependability evaluation, have been extended and also used to evaluate certain security metrics. In a number of publications, state-based stochastic models, like Markovian models or stochastic Petri nets, have been introduced as useful tools for quantitative security analysis. On the other hand, probabilistic models have also extensively been used to evaluate security metrics. For instance, some models such as attack graphs, Bayesian networks and model checking can probabilistically be defined and analyzed. In contrast to stochastic models, which are parameterized with continuous probability distributions, these types of models are primarily parameterized with discrete probability distributions.

Sallhammar et al. [1] suggest the use of game theory as a method for computing the probabilities of expected attacker behavior in a quantitative stochastic model of security. By viewing system states as elements in a stochastic game, they compute the probabilities of expected attacker behavior and model attacks as transitions between the system states. Having solved the game, the expected attacker behavior is reflected in the transitions between the states in the system model, by weighting the transition rates according to probability distributions. The proposed game model is based on a reward and cost concept and a detailed evaluation of how the reward- and cost parameter influence the expected attacker behavior is included. In the final step, continuous-time Markov chain (CTMC) is used to compute operational metrics of the system. In [2], Madan et al. used the state transition model of the scalable intrusion-tolerant architecture (SITAR) proposed in [3] and stochastic modeling techniques to capture the attacker behavior as well as the system's response to a security intrusion. The security quantification analysis is first carried out for steady-state behavior leading to metrics like steady-state availability. By transforming this model into a model with absorbing states, they computed a security metric called the mean time to security failure (MTTSF) and also the probabilities of security failure due to violations of different security attributes. Wang et al. [4] developed a stochastic reward net (SRN) model to capture attacker behavior as well as system response for the intrusion-tolerant system named SITAR. It was shown that the resulting analysis is useful in determining gains in security by reconfiguring such a system in terms of increase in redundancy under varying threat levels. Stevens et al. [5] described a probabilistic model for validating an intrusion-tolerant system that combines intrusion tolerance and security. The models were built with stochastic activity networks (SANs) formalism using the Möbius tool. This paper illustrates how probabilistic modeling can be used in an integrated validation procedure and successfully brings insight and feedback to a design. Also, Singh et al. [6] proposed an approach by using SANs to quantitatively validate an intrusion-tolerant replication management system. For this purpose, they characterized the intrusion tolerance provided by the system through several metrics defined on the model and studied the variations in these metrics in response to changes in system parameters to evaluate the relative merits of various design choices.

In [7], Kaâniche et al. presented some empirical analyses based on the data collected from the *Leurré.com* honeypot platforms deployed on the Internet. They provided some preliminary statistical modeling studies such as the analysis of the time evolution of the number of attacks taking into account the geographic location of the attacking machine, the characterization and statistical modeling of the times between attacks and the analysis of the propagation of attacks throughout the honeypot platforms in order to characterize attack processes. In [8], Jonsson et al. worked out a hypothesis on typical attacker behavior based on empirical data collected from intrusion experiments; attacking process can be split into three phases: the learning phase, the standard attack phase, and the innovative attack phase. The collected data indicated that the times between breaches during the standard attack phase are exponentially distributed. Oratalo et al. [9] provided the results of an experiment of security evaluation. The evaluation is based on a theoretical model called the privilege graph and transformed into a Markov model, which describes the system vulnerabilities that may offer opportunities to potential attackers to defeat some security objectives. They studied several modeling assumptions and discussed the validity of these assumptions based on an experimental study performed on a real system during more than a year.

McQueen et al. [10] proposed a new model for estimating the time to compromise of a system component that is visible to an attacker. The model provides an estimate of the expected value of the time-to-compromise as a function of known and visible vulnerabilities, and attacker skill level. The time-to-compromise random process model is a composite of three subprocesses associated with attacker actions aimed at the exploitation of vulnerabilities. As another example, Leversage et al. [11] proposed a mean time-to-compromise interval as an estimate of the time it would take for an attacker with a specific skill level to successfully impact a target system and a state space model and algorithms for estimating attack paths and state times to calculate these intervals for a given target system. For estimating state times, they have used a statistical algorithm based on a modified variant of McQueen' time-to-compromise model. Furthermore, Paulauskas et al. [12] proposed a method for security evaluation by using mean time-to-compromise criteria. They postulated that normal distribution should be used for mean time-to-compromise evaluation in the attacker skill level group instead of top attacker skill values, because the beginners initiate a greater number of attacks.

Xu et al. [13] proposed a stochastic model for quantifying security of networked systems. For this purpose, a vulnerability graph is used to abstract a networked system and a stochastic process used to describe attacks over the vulnerability graph. They investigate the problem from a high-level abstraction, which also leads to both analytical results and practical methods for obtaining the desired security quantities. Basagiannis et al. [14] introduced probabilistic model checking as an efficient tool-assisted approach for systematically quantifying denial of service (DoS) security threats. In order to do so, a security protocol with a fixed network topology using probabilistic specifications for the protocol participants is modeled and a probabilistic attacker model which performs DoS-related actions with assigned cost values attached into the protocol model. From the developed model, a discrete-time Markov chain (DTMC) via property preserving discrete-time semantics is obtained. The DTMC model is verified using the PRISM model checker that produces probabilistic estimates for the analyzed DoS threat.

In [15], Bodei et al. have used a special process algebra for predicting quantitative metrics such as the costs of successful attacks on systems describing cryptographic protocols. They have described protocols and their attacks using the process algebra and have associated a cost with each transition in the transition system. The transition system is then mapped to a CTMC and the performance of system is evaluated. Ahmed et al. [16] proposed a novel security metric framework that identifies and quantifies objectively the most significant security risk factors, which include existing vulnerabilities, historical trend of vulnerability of the remotely accessible services, prediction of potential vulnerabilities for any general network service and their estimated severity and finally policy resistance to attack propagation within the network. Using real life vulnerability data of the past six years from national vulnerability database (NVD), they validated this hypothesis that if a service has a highly vulnerability-prone history, then there is higher probability that the service will become vulnerable again in near future. In [17], Cervesato has investigated intruder models that rely on capabilities beyond Dolev-Yao gentlemen correctness. This quantitative approach enables evaluating protocol resilience to various forms of DoS, guessing attacks and resource limitation. While the methodology is general, it is demonstrated through a low-level variant of the multi-set rewriting (MSR) crypto-protocol specification language.

Hecker et al. [18] present two principally possible and distinct approaches to an operational security assurance evaluation of networked IT systems and discuss their pros, cons and limits, illustrated through examples. Their analysis clearly distinguishes security assurance from related subjects like security, dependability and trust. In [19], Jafari et al. propose an approach for developing security metrics to be used for assessing security posture of healthcare organizations. The metrics for this approach shall not be tailored to any specific organization to ensure comparable results. The approach is geared to articulate and visualize healthcare specific security concerns to enable stakeholders in this field to grasp an understanding of the security of their systems. The aim is to foster confidence in sharing sensitive patients' information.

Wang et al. [20] propose an attack graph-based probabilistic metric for network security and study its efficient computation. They first define the basic metric and provide an intuitive and meaningful interpretation to the metric. They then study the definition in more complex attack graphs with cycles and extend the definition accordingly. They show that computing the metric directly from its definition is not efficient in many cases and propose heuristics to improve the efficiency of such computation. Boyer et al. [21] proposed a specific set of technical security metrics for use by the operators of control systems. Their proposed metrics are based on seven security ideals associated with seven corresponding abstract dimensions of security. They have defined at least one metric for each of the seven ideals. Each metric is a measure of how nearly the associated ideal has been achieved. These seven ideals provide a useful structure for further metrics development. A case study shows how the proposed metrics can be applied to an operational control system. Lemay et al. [22] formally defined the ADversaryVIew Security Evaluation (ADVISE) method. Their approach is to create an executable state-based security model of a system and an adversary that represents how the adversary is likely to attack the system and the results of such an attack. The attack decision function uses information about adversary attack preferences and possible attacks against the system to mimic how the adversary selects the most attractive next attack step. The adversary's decision involves looking ahead some number of attack steps. System architects can use ADVISE to compare the security strength of system architecture variants and analyze the threats posed by different adversaries.

Lippmann et al. [23] have introduced meaningful security metrics that motivate effective improvements in network security. They present a methodology for directly deriving security metrics from realistic mathematical models of adversarial behaviors and systems and also a maturity model to guide the adoption and use of these metrics. Four security metrics are described that assess the risk from prevalent network threats. These initial four metrics and additional ones which are developed should be added incrementally to a network to gradually improve overall security as scores drop to acceptable levels and the risks from associated cyber threats are mitigated. Manadhata et al. [24] formalize the notion of a system's attack surface and use a system's attack surface as an indicator of the system's security. Intuitively, a system's attack surface is the set of ways in which an adversary can enter the system and potentially cause damage. Finally, Moayedi et al. [25] introduce a novel approach to extend the basic ideas of applying game theory in stochastic modeling. The proposed method classifies the community of hackers based on two main criteria used widely in hacker classifications, which are motivation and skill. They use Markov chains to model the system and compute the transition rates between the states based on the preferences and the skill distributions of hacker classes. The resulting Markov chains can be solved to obtain the desired security measures.

The studies reviewed above suggest that stochastic and probabilistic modeling techniques can be considered as feasible tools for modeling attack process and quantifying security metrics. In this paper, following the past research, we want to evaluate security metrics from a slightly new perspective. We propose a general approach with the intention of the assessment of a network's security from an information-theoretic perspective. One of the advantages of this type of model is its strong mathematical aspect. The mathematical basis of information theory is probability theory.

Therefore, it makes it possible to build a probabilistic model for the security analysis.

3. The proposed security analysis approach

In this section, we present the details of our proposed approach.

The principal goal of this paper is to propose a quantitative metric for security evaluation: Mean Privacy. In a nutshell, we propose an information-theoretic model for quantifying the security level of systems. Our approach for performing a quantitative security analysis is as follows. First of all, we define the desirable metric. For this purpose, we propose a definition of privacy of computer network. We then consider a suitable model of attack process and parameterize it with discrete distribution functions. Attack trees are known as useful models in security analyses. In order to parameterize the attack tree, we use discrete probability distribution functions. By doing so, this parameterized tree can be effectively viewed as a probabilistic model. Finally, we develop a systematic method for the evaluation of the metric. For analyzing this model, a recurrence formula will be developed. We will provide both the intuitive interpretation and the formal proof of the correctness of the proposed model.

3.1. A security metric

The aim is to define a security metric for quantitative analysis of systems. The idea behind the definition of this metric can be simply expressed as follows. In an attack process, there are a number of sources of uncertainty due to the diversity of the attacks. In other words, in order to perform a rigorous security analysis, the dynamics of all the probable attack plans must be captured. Thus, we want to introduce a methodology in order to be able to quantify our intuitive interpretation of how attackers act and how much they are predictable. For performing the attack process successfully, the attackers need to obtain necessary information about the system. In fact, the success of the attackers depends on the amount of the information that can be collected. The "private information" can be any type of useful information, which will be helpful for performing the attacks. On one hand, the general information about a typical system (e.g., system size and topology, services, open ports, etc.) can be considered as useful information. On the other hand, security vulnerabilities are also considered as appropriate information. For our purposes, we propose the following definition of a computer system's privacy:

> *Privacy of a computer system from attackers' perspective means that the attackers, to conduct a successful attack process, cannot obtain sufficient private information about the system.*

In fact, the privacy of a computer system shows how much the system would be attackable. The amount of the privacy of a computer system can be a good estimation of the extent of the capabilities of attackers. It should be noted that in real situations, attackers can get some information about systems. Therefore, it is reasonable to investigate the privacy of systems from a relative point of view. It is desirable to define a metric that can estimate how much attack scenarios that aim at violating the security of a system are predictable. The metric of interest is defined as follows:

> *Mean privacy is a metric that quantifies the amount of private information that can be obtained by attackers about a computer system.*

This metric is an indicator of the privacy level of computer system, that is, the diversity of malicious attacks. Hence, the larger the metric, the more insecure the system. It should be noted that we prefer to call the metric "mean privacy" of a computer system, because it is a weighted average of uncertainty of different attack scenarios. As will be shown, it is a comparative security metric between two networks and shows a network's security strength in units of bits. For various versions of systems operating in a variety of environments, it is useful to compare their security. Its value is nonnegative and becomes zero if we have a completely secure system. Although, in practice, a metric of a zero value cannot be achieved, it would be desirable to reduce its value as much as possible. With this definition of mean privacy in mind, in many situations, the amount of the mean privacy associated to a computer system under a specific attack process is actually counterintuitive. It means that the trust to the intuition may give rise to a wrong interpretation of the security level of the system. Additionally, the quantification of the intuition makes it possible to determine how much our assumptions about the complexity of attacks are far from actual situation.

3.2. Attack process modeling

In order to get a comprehensive view of all actions that attackers will be able to conduct so as to transfer the system under attack into an insecure state, an attack model needs to take all categories of possible attacks into account. In fact, in order to perform a security analysis of a system, two questions need to be answered: From where do attackers mainly attack (what are sources of attacks)? How do attackers perform attacks (what are attack scenarios)? Ideally, an appropriate model of an attack process needs to provide the answers of the two questions. In this regard, attack trees can be exploited. This type of attack model can incorporate different types of attacks into a unified framework and tell us the answers to the two questions. Of course, we intend to express the answers quantitatively. Depending on the applications of an attack model, it is necessary to consider an appropriate abstraction level in representing attack patterns. Our aim is to construct a high-level graphical attack model. In different situations of security analysis, and in particular, when we want to derive quantitative metrics, the different attributes of attacks make security analysis largely difficult. The main advantage of our attack modeling method is that it provides us the possibility of studying attacks with different attributes. As a result, we can propose a unifying framework for analyzing all kinds of attacks. In the proposed model, the complete details of the behavior of attackers are not necessary. In fact, by providing high-level descriptions of attacks, it would be feasible to cover different categories of possible attacks. The construction of an attack tree for describing an attack process against a system is carried out with respect to the results of vulnerability analysis of the system as well as the nature of interaction environment of the system with its users. What is usually assumed in a security analysis is that attackers remotely, without physical access to systems, launch their attacks in order to penetrate the systems. The big advantage of the model is the possibility of taking all types of attacks such as cyber, physical and social engineering into account.

Now let us go into detail about attack trees. An attack tree is simply a rooted tree that is used in the context of security analysis of systems (e.g., see [26–32]). It can be exploited to model systematically all possible attacks against a system and to analyze the system security from various perspectives. An attack tree represents a collection of attack scenarios starting from its leaves and ending in the root. That is, the root of the attack tree denotes security violation state. It means that the final goal of the underlying attack process is characterized by the root of the attack tree. Intermediate goals (i.e., subgoals) are characterized by the internal nodes of the attack tree. Each internal node of the attack tree represents a subgoal of the attack process. The leaves, at the lowest level, are simply atomic attacks, which can no longer be decomposed and extended. They show different ways to reach the root of the attack tree.

Generally, traditional attack trees can consist of two types of internal nodes: AND nodes and OR nodes. An OR node characterizes an attack goal that will be achieved if any one of its subgoals can be achieved. An AND node characterizes an attack goal that will be achieved only if each and every of its subgoals can be achieved. Here, for our needs, we also consider a variant of tree structure with both types of AND nodes and OR nodes. An example of a typical attack tree is given in Fig. 1. It should be noted that although we use the structure of the traditional attack tree models, the nature of the evaluation method of our model is primarily different from that of the traditional attack trees.

As stated the above, in an attack tree, root, internal nodes and leaves are supposed to represent the final goal, intermediate goals and atomic attacks, respectively. As an alternative interpretation, the nodes represent attackers' privilege levels and the edges represent attack actions for increasing current privilege level. In this paper, we consider a different method of the evaluation of attack tree structures for our needs. From the structure point of view, the root is also supposed to be the security goal of the attack process. Also, at the lower levels of the tree, for launching the attack process, intermediate nodes and leaves are respectively considered as the intermediate subgoals and starting points. Usually, from the evaluation method point of view, in the context of quantitative security analysis using attack trees, discrete or continuous values are assigned to leaves of attack trees. Once the model has been parameterized, the security analysis can be performed. The result metric of the analysis is determined by the nature of the values assigned to the leaves of the attack tree. Here, conversely, suitable numerical values are assigned to all edges of the tree structure. After creating the attack tree structure, the values are assigned to each edge of the tree so that the assigned value to each edge represents the probability of choosing the corresponding branch. In fact, for each node, we define a discrete random variable specified by a probability mass function. In other words, for each goal, its corresponding random variable, based on a discrete distribution, covers all of its possible subgoals.

After constructing an attack tree model of an attack process, it would be feasible to evaluate some security metrics. For this purpose, the leaves of the attack tree are initialized by labeling with special values, depending on the kind of the security metric that is desirable to be evaluated. For the quantitative analysis using an attack tree, it is attempted to suitably incorporate probabilistic values into the attack tree. For a specific node, let X be its assigned random variable that takes on values in a finite set like $\{x_1, x_2, \ldots, x_n\}$. Each of the values $x_1, x_2, \ldots, x_n$ denotes a possible branch of the node. The assignment is illustrated in Fig. 2. Note that, for the evaluation process, we assign only one parameter to attack tree. In most cases, we usually get access to a limited amount of information about the characteristics of attackers. Thus, it is desirable to be able to predict the behavior of attackers even if we have the limited amount of information in hand.

3.3. Quantitative evaluation

It is time to discuss the process of the quantitative evaluation of the metric. As stated earlier, we intend to propose an information-theoretic model. Thus, it is necessary to clarify what is actually our interpretation of an information-theoretic perspective. Broadly speaking, information theory presents the concepts and methods that are applicable to the quantification of information. Here, the aim is to adopt a suitable interpretation of information, which is compatible with information of concern in the area of security analysis. From an attacker's point of view, information of particular importance is the information that is required to conduct attacks against computer systems. If we are able to define the concept of information as stated, we can expect to get an information-theoretic model for quantifying the security level of computer systems. For doing so, we utilize the concept of entropy. Entropy is a quantity that is used widely in the field of information theory. It is a quantity of ultimate important that is used for the measurement of uncertainty of a random variable. Let X be a discrete random variable with a finite number of values $x_1, x_2, \ldots, x_n$ that their corresponding probability values are $p_1, p_2, \ldots, p_n$, respectively. The entropy of the discrete random variable X with the probability mass function $p_X(x_i) = P(X = x_i) = p_i$, for $1 \leqslant i \leqslant n$, is defined to be:

$$H(X) = -\sum_{i=1}^{n} p_i \log_2 p_i.$$

For a simple proof of why this definition of entropy works well, you can refer to [33]. It should be noted that for the computation of entropy, the different values of X are not taken into account, and in fact, it only depends on the probability distribution of Y. Also, note that since the logarithm function is to base 2, then the entropy value is being measured in unit of bits. In fact, it is the average number of bits required to represent the random variable X. Although, entropy can be defined for both discrete and continuous types of random variables, for our purposes, we consider only the case of discrete.

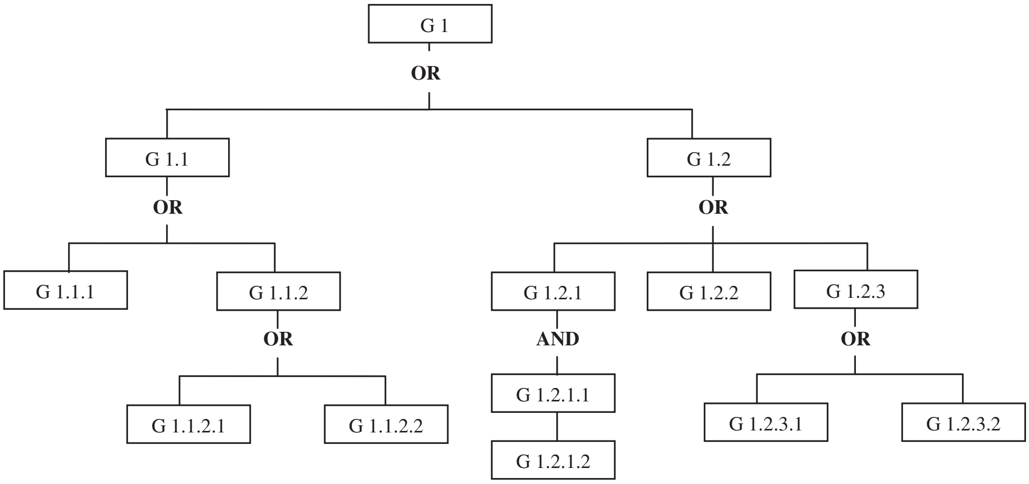
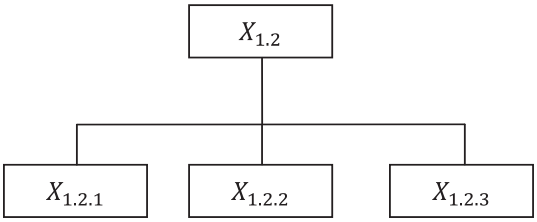


Fig. 1. Example of a typical attack tree.



$X_{1.2}$	$x_{1.2.1}$	$x_{1.2.2}$	$x_{1.2.3}$
$P_{X_{1.2}}(x)$	p_1	p_2	p_3

Fig. 2. The assignment of a probability mass function to the node G1.2 of the attack tree in Fig. 1.

– Entropy is defined to be the amount of uncertainty of a random variable. In fact, it quantifies the amount of unknown information that can be extracted from the random variable. Thus, with an appropriate extension, it can be effectively regarded as a quantitative indicator that may be applicable in the context of security analysis. In order to utilize the entropy function in the context of security analysis, we first have to show why this function can satisfy the desirable properties needed for the description of the uncertainty in the behavior of attackers. It has some interesting properties that are much in agreement with our purposes for characterizing the behavior of attackers. Entropy functions are themselves defined as functions of probability mass functions. We first specify the basic properties of entropy function and then show why they are well fitted to be applied to carrying out the quantitative evaluation of the security of computer systems by capturing the attacker behavior: *It quantifies the uncertainty of a random variable.* Let Y be a discrete random variable with finite possible values $\{y_1, y_2, \ldots, y_k\}$. Also, let $p_Y(y_i) = P(Y = y_i) = p_i$, for $1 \leqslant i \leqslant k$, be the probability mass function of the random variable Y. The entropy of the random variable Y is defined to be:

$$H(Y) = -\sum_{i=1}^{k} p_i \log_2 p_i.$$

What does this imply for security analysis? The main application of entropy is to express the intuitive understanding of the complexity of attacks as a measurable concept. Accordingly, the dynamics of actions of attackers can be described by using the definition of a set of discrete random variables and the computation of their respective entropy. In other words, it is possible to quantify the amount of uncertainty in the behavior of attackers based on the values of entropies of the random variables. As will be seen, we develop a hierarchical view for establishing relationships among the random variables.

– *It is a nonnegative quantity and takes its maximum value if and only if its underlying probability distribution is a discrete uniform distribution.* Let Y be the above random variable. We have:

$$0 \leqslant H(Y) \leqslant \log_2 k,$$

with equality if and only if Y has a uniform distribution over $\{y_1, y_2, \ldots, y_k\}$.

What does this imply for security analysis? This property of the entropy function is simply consistent with the behavior of typical attackers. During an attack process, attackers have some choices that each of them is made with a corresponding probability value. From the attack complexity point of view, the worst case scenario occurs when all of these choices are equally likely to be made. The reason of this is that we cannot assume higher probabilities for specified actions of the attackers, due to the lack of sufficient information. But if we know that a number of the actions have higher probabilities compared to the others, then the amount of the uncertainty naturally decreases.

– *It is an increasing function in the number of its arguments.* Let Y and Z be two discrete random variables having uniform distributions with finite possible values $\{y_1, y_2, \ldots, y_n\}$ and $\{z_1, z_2, \ldots, z_m\}$, respectively. In other words, let $p_Y(y_i) = P(Y = y_i) = \frac{1}{n}$, for $1 \leqslant i \leqslant n$, and $p_Z(z_i) = P(Z = z_i) = \frac{1}{m}$, for $1 \leqslant i \leqslant m$, be the probability mass functions of the random variables Y and Z, respectively. If $n \leqslant m$, then:

$$H(Y) = \log_2 n \leqslant \log_2 m = H(Z).$$

What does this imply for security analysis? Intuitively, as the diversity of attacks increases, the amount of the corresponding entropy function also increases. On one hand, different classes of attackers have their own skill levels, resources and objectives. On the other hand, a typical computer system has a number of vulnerabilities. With respect to these attributes, the attackers are likely to perform a variety of attack actions against the network. These actions characterize different ways to get a security goal. Clearly, in situations where there are more possibilities of actions for the attackers, the amount of the uncertainty in the selections of the attackers will increase. In line with a number of works (e.g., see [34,35]), we also believe that the field of information theory can be applied, with good extension, as an effective means in the context of security analysis.

Now, let us develop a systematic method for the evaluation of the metric. Since entropy is defined for random variables, in order to utilize the concept of entropy in the area of security assessment, our first task will be to define a set of appropriate random variables. These random variables need to be defined so that they can capture the diversity of attacks. Since we are not interested in continuous attributes of the attacker behavior (e.g., duration times of attacks) and our desirable attribute is the diversity of attacks in an predictable attack process, the defined random variables need to be of discrete types, each of them with a finite number of possible values.

Due to the special structure of an attack tree, it is possible to obtain a recurrence relation for computing the uncertainty of the whole attack tree. In general, an attack tree structure can consist of a number of smaller attack tree structures, called attack subtrees, and its respective metric can be computed as a function of these smaller structures. As a result, it would be possible to systematically derive the final formula for the evaluation of the metric.

As discussed before, for our purposes, we follow a new paradigm for the parameterization of attack trees. The objective is to quantify our intuition of the amount of uncertainty of an attack process that is described by an attack tree model. Naturally, the uncertainty associated with an attack tree increases if the branching factor of its internal nodes and the number of its levels increase. Hence, these are considered as effective aspects in increasing the uncertainty of attack trees. Of course, they are themselves strictly dependent on the probability distributions assigned to the nodes of attack trees. For each internal node of the tree, a discrete random variable is defined and the entropy of this random variable will be computed. The definition of such a random variable is carried out as follows. The possible values of the random variable actually represent the possible selections of attack patterns and their corresponding probabilities are the selection probabilities of these patterns.

In summary, the overall process of the evaluation of the metric consists of two “static” and “dynamic” phases. In the static phase, which is actually the initialization phase, each internal node is assigned a discrete probability mass function. Based on these distributions, the initial entropy values of all the internal nodes are computed. Clearly, the initial values of the uncertainty of the leaves are set to zero.

The details of the static phase are as follows.

3.3.1. Static phase

Generally, an attack tree can have many leaves. Clearly, the branches of the tree are completely terminated at these leaves (e.g., see the nodes G1.1.2.2 and G1.2.3.2 of the attack tree in Fig. 1). Accordingly, there is no uncertainty when being in a leaf node. Consequently, it is reasonable to assume that the entropy of a leaf node has an initial value, – which is also its final value – equal to zero. This assumption will be considered to be the initial condition for the recurrence relation defined for the estimation of the quantitative metric for the whole attack tree. To be able to express this assumption formally, we can define some dummy random variables for the leaves. Initially, the uncertainty values of all internal nodes are independently computed. Also, the values of uncertainty of the leaves are set to zero. Note that leaves are at the lowest level of an attack tree, and at this level, further branches do not occur. The value of the uncertainty of an internal node is a function of the probability distribution, which is distributed over its edges, and the values of uncertainty of all its children nodes. For a specific leaf node l, let X_l be the dummy random variable that corresponds to it. Thus, by assumption we have:

$$H(X_l) = 0.$$

Now, we consider an arbitrary internal node I of a general attack tree. It is supposed that this node has $m \geqslant 2$ children; in addition, each of its children may be itself either an internal node or a leaf (e.g., see the nodes G1.2 and G1.1.2 of the attack tree in Fig. 1). In such situation, there are m possibilities of selecting a subgoal. For the specific internal node I, let X_I be the dummy random variable that corresponds to it. In addition, let $p_{X_I}(x_i) = P(X_I = x_i) = p_i$, for $1 \leqslant i \leqslant m$, be the probability mass function of the random variable X_I. Therefore, we can evaluate the static uncertainty of such nodes as follows:

$$H(X_I) = H(p_1, p_2, \ldots, p_m) = -\sum_{i=1}^{m} p_i \log_2 p_i.$$

Here, two special cases are of interest. One special case is when the internal node has only one child. This case occurs whenever the node is either an OR node with only one child or an AND node. It means that there is only one choice for attackers, and thus, there is no uncertainty when being in such a node. That is, the initial uncertainty of the node is clearly computed to be zero:

$$H(X_I) = H(0, 0, \ldots, 1, \ldots, 0) = 1 * \log_2 1 = 0.$$

Another special case is when the random variable of the node obeys a discrete uniform distribution. This case occurs whenever all the attack subgoals are equally likely to be selected. If the distribution of the node is uniform, then for all $1 \leqslant i \leqslant m$, we have $p_i = \frac{1}{m}$. Thus, the uncertainty of the node will reach its maximum value:

$$H(X_I) = H\left(\frac{1}{m}, \frac{1}{m}, \ldots, \frac{1}{m}, \ldots, \frac{1}{m}\right) = -\sum_{i=1}^{m} \frac{1}{m} \log_2 \frac{1}{m} = \log_2 m.$$

In the static phase, the initial uncertainties are computed for all the nodes. In the dynamic phase, the final uncertainties will be computed for all the nodes. In the dynamic phase, the current values of the nodes are propagated, one level at a time, up to their parent nodes. The details of the dynamic phase are as follows.

3.3.2. Dynamic phase

Here, for our analysis, we consider a typical attack tree model of an attack process. Because of the hierarchical structure of attack trees, it is possible to develop a recursive algorithm for computing the metric of interest. In our analysis, we assume that the structure of the attack tree would be fixed and the only difference would be in defining probability distributions. Depending on the probability distributions assigned to the nodes of this fixed attack tree, different values for the metric can be obtained. In some situations, due to the increased knowledge of some system vulnerabilities, the distributions of the nodes are assumed to be less uniform. The correct choice of probability mass functions to describe the diversity of attacks by the model is one of the most important challenges in evaluating a reliable metric. Therefore, for a specific attack tree structure, the values of the metric can differ considerably for probability distributions of the same structure, leading to “worst case”, “best case” and “average case” scenarios. In practice, what is important is the average case scenario. Hence, we seek to find a recurrence relation for evaluating the metric in the average case. Accordingly, we call the metric “mean privacy”.

An attack tree, in any form, has only a root. It is this node at which the security actions will be successfully terminated. Naturally, it is supposed that the original tree cannot be empty. Let us consider the general tree AT structure that is shown as in Fig. 3. As can be seen, here, the root has n children. Each of these n nodes may be itself the root of a smaller attack tree. Of course, we note that some of these subtrees may be trees with only one node; that is, their roots – which are the only existing node of them – are leaves of the original tree. Due to the recursive structure of the attack tree, the attack subtrees can be treated themselves as general attack trees.

To be able to carry out the required computations, we define the discrete random variable Y as follows. It is defined to characterize the diversity of the subtrees of the root. The set of all values of Y is assumed to be $\{y_1, y_2, \ldots, y_n\}$. Let the ith branch occur with the probability p_i. It means that with the probability p_i, the attack process will be described by the ith attack subtree. Thus, we define the probability mass function of the random variable Y as follows:

$$p_Y(y_i) = P(Y = y_i) = P\{\text{attack subtree } AT_i \text{ is chosen}\} = p_i, \quad \text{for } 1 \leqslant i \leqslant n.$$

Also, consider the vector X to be defined to characterize the type of the subtrees of the root. In other words X is simply a vector of “attack subtrees”:

$$X = (AT_1, AT_2, \ldots, AT_n).$$

Depending on each of the different values of Y, the appropriate component of the vector X will be utilized to quantify the uncertainty of the specified subtree:

$$X|Y = y_i \equiv X_i.$$

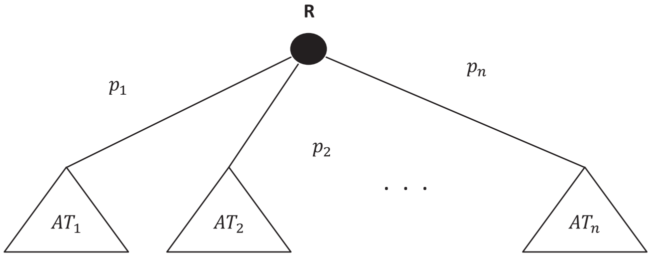


Fig. 3. The recursive structure of an attack tree.

Intuitively, the uncertainty of the general attack tree AT is effectively a function of (1) the initial uncertainty of the root R and (2) the uncertainty of n smaller attack trees:

$$MP(AT) = f(R, X) = f(p_1, p_2, \ldots, p_n, AT_1, AT_2, \ldots, AT_n).$$

In order to evaluate the uncertainty of the general attack tree, we first need to evaluate the initial uncertainty of the root R, as done in the static phase:

$$H(R) = H(Y) = H(p_1, p_2, \ldots, p_n) = -\sum_{i=1}^{n} p_i \log_2 p_i.$$

Now, we need to investigate how to compute the uncertainty of the attack subtrees. Suppose that the uncertainty of the general attack tree AT is equal to $MP(AT)$ and let the uncertainty of the ith attack subtree AT_i be $MP(AT_i)$. The calculation of $MP(AT_i)$ is effectively performed in a similar way to that of $MP(AT)$. The only difference is that the quantity $MP(AT_i)$ is calculated for a subtree of the original tree. Each of such attack subtrees has fewer attributes such as the number of nodes and branches compared to those of the attack tree. Since this attack subtree is selected with some probability less than unity, it actually produces a fraction of the total uncertainty of the attack process. Formally, the actual proportion of the uncertainty (i.e., the effective uncertainty) of the ith attack subtree needs to be calculated as follows:

$$p_i \times MP(AT_i).$$

For taking all the subtrees into account we take the advantage of the concept of mathematical expectation. Let T be a discrete random variable with finite possible values $\{MP_1, MP_2, \ldots, MP_n\}$, where $MP_i = MP(AT_i)$. Also, let $p_T(MP_i) = P(T = MP_i) = p_i$, for $1 \leqslant i \leqslant n$, be the probability mass function of the random variable T. The expectation of the random variable T is given by:

$$E[T] = \sum_{i=1}^{n} MP_i p_T(MP_i) = \sum_{i=1}^{n} MP(AT_i) p_i.$$

It should be clear from the discussion that the metric of interest, associated with the general attack tree, can be simply obtained by considering all effective factors that have impact on the uncertainty of the attack process. The first reason of the uncertainty is due to the branching operation at the root. At the next levels, there are n attack subtrees, which are the other sources of the uncertainty. Thus, the metric representing the complexity level of the attack process will be a function of the initial uncertainty of the root and the expected uncertainty of the attack subtrees. Taking these factors into account, it will be possible to develop a recurrence relation to compute the metric for the general attack tree:

$$MP(AT) = H(R) + \sum_{i=1}^{n} MP(AT_i) p_i,$$

which can also be rewritten as follows:

$$MP(AT) = -\sum_{i=1}^{n} p_i \log_2 p_i + (p_1(MP(AT_1))) + \cdots + (p_i(MP(AT_i))) + \cdots + (p_n(MP(AT_n))).$$

This recurrence relation is used to evaluate the security metric. Since each of the smaller attack trees may be itself a general attack tree, it is necessary to continue the evaluation process until the stopping condition will be reached. Notice that the stopping condition is reached when leaves are visited. Thus, the initial condition of the relation is as follows:

$$MP(l) = H(X_l) = 0,$$

where l is a leaf and X_l is its respective random variable.

This general formula can give us some reasonable outcomes. For example, as a special case of an attack tree, if the attack tree has only one attack subtree, the uncertainty of the attack tree is equal to that of the subtree; in such situation, there is no branching, and hence, we have:

$$MP(AT) = MP(AT_1).$$

As another case, consider an attack tree consisting of only one attack path (see Fig. 4). Such an attack tree has exactly one starting point and one ending point. Intuitively, we have no branching and therefore, no uncertainty about the behavior of attackers:

$$MP(AT) = 0.$$

It should be noticed that the above formula has a recursive form. Using the formula, the evaluation process can be performed top-down, one level at a time. The process will be terminated when the leaves at the lowest level are visited. Thus, based on the recurrence relation, we can design a recursive algorithm for computing the metric. The pseudocode of this top-down algorithm is given in Fig. 5. From the computation point of view, we can also implement a bottom-up version of the algorithm. In other words, the metric can be evaluated from the bottom up. Starting from the leaves, the values of the nodes are propagated up, one level at a time, to their parents. This process continues until the root is reached.

3.4. Applications

Until now, we have performed the process of defining and computing the security metric of interest. Note that the defined metric is quantitative, and thus, it can be used as an appropriate indicator for the quantitative analysis of a system's security. It is possible to individually obtain this metric for different versions of a system or even for the principal components of a single system. After evaluating the metric and performing the analysis, we should try to identify and reduce security problems of the system as much as possible. It would be useful to briefly explain the potential applications of the quantitative model. The major applications of the model are as follows:

1. *Evaluating the security level of a system.* As discussed in Section 1, in practice, it is strongly accepted that an absolute system security level cannot be achieved. Instead, a relative security analysis is of interest. The quantitative analysis can be done to get a good understanding of the intensity and dynamics of potential

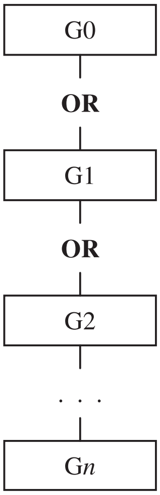


Fig. 4. Example of an attack tree representing an attack process consisting of an attack path.

```
ALGORITHM MeanPrivacy(AT )

//Computes recursively the mean privacy of an attack tree
//Input: A tree $AT$ with $n$ nodes, an array $BF[1..n]$ of branching factors for a list of $n$ nodes
//Output: The mean privacy of $AT$

// The static phase


$H[r(AT)] = 0$
If $BF[r(AT)] \neq 0$
   $n = BF[r(AT)]$
   for $i \leftarrow 1$ to $n$ do
        $H[r(AT)] \leftarrow H[r(AT)] - p_i \log_2 p_i$

// The dynamic phase


$MP = H[r(AT)]$ // The mean privacy
if $T = \emptyset$ return 0
else
     for $j \leftarrow 1$ to $BF[r(AT)]$ do
          $MP \leftarrow MP + p_j * MeanPrivacy(AT_j)$
return $MP$
```

Fig. 5. A recursive algorithm for computing the metric.

threats against a system. Consequently, the metric can guide us to decide where security attempts must be applied to increase the security level using cost-effective methods.

2. *Comparing the security level of systems.* From a relative security analysis point of view, it would be meaningful to utilize the metric as a comparative criterion between two different versions or configurations of a system, or more generally, between two or more distinct systems so as to quantify and contrast the level of their expected security. In fact, if we have several options for choosing a system, then this metric can help us to determine which version of the system is the most secure one. For this purpose, if we want to compare two systems and choose the more secure one, it is sufficient to quantify their security level.
3. *Identifying security problems.* As another application of the proposed model, it can be utilized to identify security weaknesses of systems. Clearly, the complexity of attacks usually stem from various sources of system vulnerabilities. The big advantage of the attack tree structure is that it can be utilized to analyze only some of security-critical components of a system rather than the whole system. Thus, the security level of different parts of a system can be evaluated and compared so as to identify more vulnerable parts of the system. By doing so, the system security experts can put the increased emphasis on the most security-critical components of the network. As a result of this analysis, it will be possible to apply cost-effective methods to strengthen the security of the network by reducing or removing vulnerabilities.

To summarize, the proposed model makes it possible to investigate new security policies, mechanisms, countermeasures and strategies that can be adopted and explored in order to obtain the arbitrarily tunable security levels.

4. An illustrative example

In this section, an appropriate example is given for demonstrating how, for a specific security-critical computer network, the process of attack modeling and quantitative analysis will be performed using the concepts and ideas discussed the above.

For illustrative purposes, we consider a hypothetical computer network consisting of a web server and a number of workstations. It is assumed that the principal task of the network is to store, manipulate, send and receive data. Thus, the security service is the "data confidentiality" that is expected to be provided by security mechanisms implemented for the network. We will go through the following steps:

- **Step 1.** The construction of an attack tree.
- **Step 2.** The computation of the security metric.
- **Step 3.** The quantitative security analysis of the computer network.

4.1. The attack tree

The first step of the analysis is to recognize all malicious attack patterns that are predictable to be conducted against the computer network. To begin with, it is necessary to specify the underlying assumptions of the computer network and the characteristics of its interaction environment with users (i.e., potential attackers). We base our analysis of the attacks on the following assumptions:

- Obviously, the network is needed to be connected to the Internet; it provides external attackers with the opportunity to remotely launch their attacks (e.g., "Install malware") with the purpose of compromising the network machines and thereby disclosing the confidential data.
- As a general rule, typical attackers should be regarded as real dangers. Usually, they have access to a wealth of attacking tools (e.g., packet sniffers, password crackers). Hence, conducting a large number of the attacks (e.g., "Guess password by brute-force methods") against the network is a highly automated process.
- The users are not generally familiar with security rules or may simply ignore such rules. Therefore, a number of the attacks (e.g., "Send e-mail with malicious code") may stem from the lack of sufficient experience and professional competence.
- For computing and communication purposes, the machines of the network are supposed to provide the users with necessary services for accomplishing their tasks. The machines need to

be equipped with appropriate operating systems and application programs. As a result, a number of the attacks (e.g., "Exploit server vulnerabilities") would be predictable.

- In addition to the cyber attacks by the external attackers, insiders (e.g., disgruntled employees) may be the potential sources of serious threats to the network security. With the possibility of the direct physical access to resources of the network and being aware of some vulnerabilities of the network, the insiders will be able to perform special kinds of physical attacks (e.g., "Steal laptops") that are not feasible from outside zone of the network.
- The confidential data are either stored on the server or transferred over communication links of the network. In other words, the attackers can utilize two principal methods to get access to the confidential data: system-level attacks and link-level attacks. Thus, they can take advantage of a number of the attacks (e.g., "Exploit buffer overflow vulnerability") to obtain privileged access to the server. Alternatively, they can also get access to the confidential data using eavesdropping techniques. These attacks (e.g., "Hijack user's session on links") are able to capture the network traffic that is exchanged over the channels between clients and servers.

Up to now, we have mentioned the underlying assumptions of the attack process. Now, after identifying the attack patterns, they need to be represented using a corresponding attack tree model. We construct an attack tree consisting of all of the predictable attack scenarios that may be used for the disclosure of the confidential data from the network under study. For the sake of the compact representation of this large attack tree, we represent it textually, as in the left-most column of Table 1. The root of the attack tree – the node that is labeled 1 – represents the final goal of the attackers; that is, getting access to the confidential data.

4.2. The quantitative evaluation of the metric

In this step, after the construction of the attack tree model, it is the time to use some estimated input data and compute the quantitative security metric called "Mean Privacy of Computer Network". It should be noted that our primary focus is on developing a methodology for doing quantitative analysis of security of computer systems. In fact, the main concern is the correctness and applicability of the proposed model, and therefore, the carefully estimated numerical values are used to clearly demonstrate how the probabilistic model can be used to obtain the metric. Here, the input parameters of the model are some probability mass functions assigned to the nodes of the tree and the output of the model is the desired security metric. Therefore, in order to evaluate the metric, we need to assign a number of probability mass functions to the nodes of the tree. They are used to indicate the selection probabilities of the attack patterns. As characterized in the left-most column of Table 1, for each internal node of the tree, we define a discrete random variable that is described by a respective probability mass function. The parenthesized number of each attack scenario indicates its selection probability.

For example, the respective random variable of the root "1. Disclose confidential data" is X_1 that its probability mass function is specified as follows:

$$P(X_1 = x_{1.1}) = 0.3, \quad P(X_1 = x_{1.2}) = 0.7.$$

It means that for disclosing the confidential data, the attackers get the data from the server directly with probability 0.3 and exploit the authorized users with probability 0.7.

As another example, for the node "1.2.1 Eavesdrop on communication channels", we have the following distribution:

$$P(X_{1.2.1} = x_{1.2.1.1}) = 0.1, \quad P(X_{1.2.1} = x_{1.2.1.2}) = 0.2, \quad P(X_{1.2.1} = x_{1.2.1.3}) = 0.2,$$

$$P(X_{1.2.1} = x_{1.2.1.4}) = 0.3, \quad P(X_{1.2.1} = x_{1.2.1.5}) = 0.2.$$

As can be seen in Table 1, the metric of interest, after the algebraic simplification will be evaluated in four steps (see the columns 2–5). Step 1 deals with the computations of the "static phase". For example, for the node "1.2.2 Take remotely control of terminals", we will obtain:

$$H(X_{1.2.2}) = -0.2\log_2 0.2 - 0.5\log_2 0.5 - 0.3\log_2 0.3 = 1.47.$$

By assumption, the initial uncertainties of the leaves are assumed to be zero. For example, for the leaf "1.2.2.3.3 Inject SQL statements", we will have:

$$H(X_{1.2.2.3.3}) = 0.$$

Steps 2 through 4 deal with the computations of the "dynamic phase". For example, for attack subtree with root "1.2 Exploit authorized users", we will obtain:

$$\begin{aligned} MP(AT_{1.2.2}) = {} & -0.4\log_2 0.4 - 0.4\log_2 0.4 - 0.1\log_2 0.1 - 0.1\log_2 0.1 \\ & + (0.4)(2.22) + (0.4)(3.06) + (0.1)(1.36) \\ & + (0.1)(0.88) = 4.06. \end{aligned}$$

Here, for simplicity, we only present the final results of the computations in the table and ignore the additional and in detail computations. It should be noted that the metric computed from the bottom up. The algorithm computes the uncertainty of the nodes level by level, starting from the leaves. The evaluation process continues until the root is reached. The value of the metric for the whole attack tree is 4.41. It is the first number in the right-most column of the table.

4.3. The quantitative analysis of the computer network

The probabilistic computations provide very important information about the diversity and complexity of the attacks. We wish to use this information to analyze the security of the computer network. The results of the analysis are utilized to improve the security posture of the network. The analysis can be done by the guidelines about the applications of the model given in Section 3.4.

First of all, let us look at the attack process in its entirety. The value of the security metric for the entire attack tree is 4.41. We want to know whether the computer network is secure or not. Since we are doing a "quantitative" analysis, we have to interpret the number from a relative point of view. In other words, we cannot talk about the security of the computer network in an absolute sense. It depends on the tolerance threshold of the network. The value tells us the complexity level of the attacks. Hence, the larger the metric, the more insecure the network. Naturally, the decreased value of the mean privacy level of the network specifies the increased security level of the network. The metric helps us to know by how we are far from absolute security. Accordingly, we would like to reduce the value as much as possible.

The metric quantifies the amount of the diversity of the attacks. It is an indicator of how much the attackers are predictable. Therefore, for the security analysis, we can compare the complexity of the attack patterns.

For example, let us compare the uncertainties of the "1.1.2.1 Install malwares" attacks and the "1.2.3 Social engineering" attacks. For this purpose, it is necessary to characterize their corresponding attack subtrees, and evaluate their uncertainties (see Fig. 6). The metrics are computed to be 1.52 and 1.36, respectively (see the right-most column of Table 1). It means that the complexity of the "1.1.2.1 Install malware" attacks is higher than that of the

Table 1
Tabular representation of the attack tree: The left-most column represents the attack tree textually, while all the other columns represent the principal steps for computing the metric. Step 1 deals with the computations of the "static phase". Steps 2 through 4 deal with the computations of the "dynamic phase". In each step, the final values are shown in bold.

	Nodes of the attack tree	Step 1	Step 2	Step 3	Step 4
1.	**Disclose confidential data (1)**	0.88	0.88	0.88	**4.41**
1.1	*Get data from server directly (0.3)*	0.97	0.97	**2.28**	**2.28**
1.1.1	Take remotely control of server (0.4)	0.88	**1.53**	**1.53**	**1.53**
1.1.1.1	Exploit server vulnerabilities (0.7)	**0.72**	**0.72**	**0.72**	**0.72**
1.1.1.1.1	Exploit operating system vulnerabilities (0.2)	**0.0**	**0.0**	**0.0**	**0.0**
1.1.1.1.2	Exploit application vulnerabilities (0.8)	**0.0**	**0.0**	**0.0**	**0.0**
1.1.1.2	Crack administrative password (0.3)	**0.47**	**0.47**	**0.47**	**0.47**
1.1.1.2.1	Guess password by brute-force methods (0.1)	**0.0**	**0.0**	**0.0**	**0.0**
1.1.1.2.2	Harvest password by packet sniffing (0.9)	**0.0**	**0.0**	**0.0**	**0.0**
1.1.2	Access server physically (0.6)	**1.17**	**1.17**	**1.17**	**1.17**
1.1.2.1	Install malware (0.7)	**0.0**	**0.0**	**0.0**	**0.0**
1.1.2.2	Get direct access confidential data (0.2)	**0.0**	**0.0**	**0.0**	**0.0**
1.1.2.3	Steal server hard drive (0.1)	**0.0**	**0.0**	**0.0**	**0.0**
1.2	*Exploit authorized user (0.7)*	1.72	1.72	**4.06**	**4.06**
1.2.1	Eavesdrop on communication channels (0.4)	**2.22**	**2.22**	**2.22**	**2.22**
1.2.1.1	Set up covert channel (0.1)	**0.0**	**0.0**	**0.0**	**0.0**
1.2.1.2	Employ pharming technique (0.2)	**0.0**	**0.0**	**0.0**	**0.0**
1.2.1.3	Employ man-in-the-middle technique (0.2)	**0.0**	**0.0**	**0.0**	**0.0**
1.2.1.4	Sniff network traffic over channels (0.3)	**0.0**	**0.0**	**0.0**	**0.0**
1.2.1.5	Hijack user's session on link (0.2)	**0.0**	**0.0**	**0.0**	**0.0**
1.2.2	Take remotely control of terminals (0.4)	1.47	**3.06**	**3.06**	**3.06**
1.2.2.1	Capture password (0.2)	**1.28**	**1.28**	**1.28**	**1.28**
1.2.2.1.1	Guess by brute force methods (0.6)	**0.0**	**0.0**	**0.0**	**0.0**
1.2.2.1.2	Install key logger (0.3)	**0.0**	**0.0**	**0.0**	**0.0**
1.2.2.1.3	Monitor user activity (0.1)	**0.0**	**0.0**	**0.0**	**0.0**
1.2.2.2	Install malwares (0.5)	**1.52**	**1.52**	**1.52**	**1.52**
1.2.2.2.1	Send email with malicious code (0.4)	**0.0**	**0.0**	**0.0**	**0.0**
1.2.2.2.2	Use hidden code (0.4)	**0.0**	**0.0**	**0.0**	**0.0**
1.2.2.2.3	Propagate through P2P network (0.2)	**0.0**	**0.0**	**0.0**	**0.0**
1.2.2.3	Exploit terminal vulnerabilities (0.3)	**1.91**	**1.91**	**1.91**	**1.91**
1.2.2.3.1	Circumvent access control mechanisms (0.4)	**0.0**	**0.0**	**0.0**	**0.0**
1.2.2.3.2	Exploit buffer overflow vulnerability (0.2)	**0.0**	**0.0**	**0.0**	**0.0**
1.2.2.3.3	Inject SQL statements (0.2)	**0.0**	**0.0**	**0.0**	**0.0**
1.2.2.3.4	Exploit cross-site scripting vulnerability (0.2)	**0.0**	**0.0**	**0.0**	**0.0**
1.2.3	Use social engineering techniques (0.1)	**1.36**	**1.36**	**1.36**	**1.36**
1.2.3.1	Take advantage of phishing web sites (0.5)	**0.0**	**0.0**	**0.0**	**0.0**
1.2.3.2	Send fake emails (0.4)	**0.0**	**0.0**	**0.0**	**0.0**
1.2.3.3	Bribe authorized user (0.1)	**0.0**	**0.0**	**0.0**	**0.0**
1.2.4	Steal terminal's physical devices (0.1)	**0.88**	**0.88**	**0.88**	**0.88**
1.2.4.1	Steal laptops (0.3)	**0.0**	**0.0**	**0.0**	**0.0**
1.2.4.2	Steal storage devices (0.7)	**0.0**	**0.0**	**0.0**	**0.0**

"1.2.3 Social engineering" attacks, while it is not clear from the structure of the subtrees whether the former is more complex than the latter.

As another example, consider the two attack subtrees as shown in Fig. 7. Visually, the upper attack subtree is more concise than the lower attack tree. Hence, it seems that the upper subtree has a lower complexity level than the other tree. However, if we compute the metrics for the two attack subtrees, which are respectively 2.22 and 1.53, (see the right-most column of Table 1) it will be concluded that the upper attack subtree actually has a higher level of complexity.

These small examples show that our intuition may give rise to the incorrect interpretations of the complexity level of the attack scenarios. In a similar vein, it is not difficult to see that the

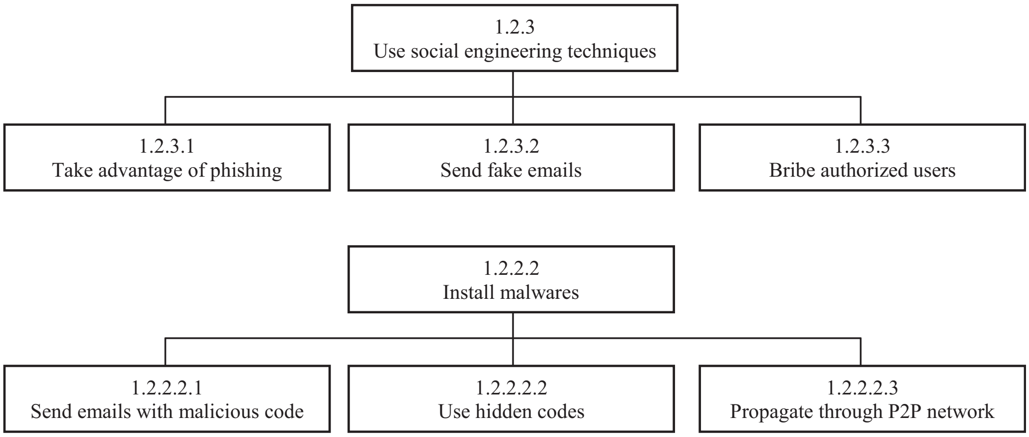


Fig. 6. The comparison of the uncertainty of two attack subtrees: (a) the uncertainty of the "Install malwares" attacks is 1.52 and (b) the uncertainty of the "Social engineering" attacks is 1.36.

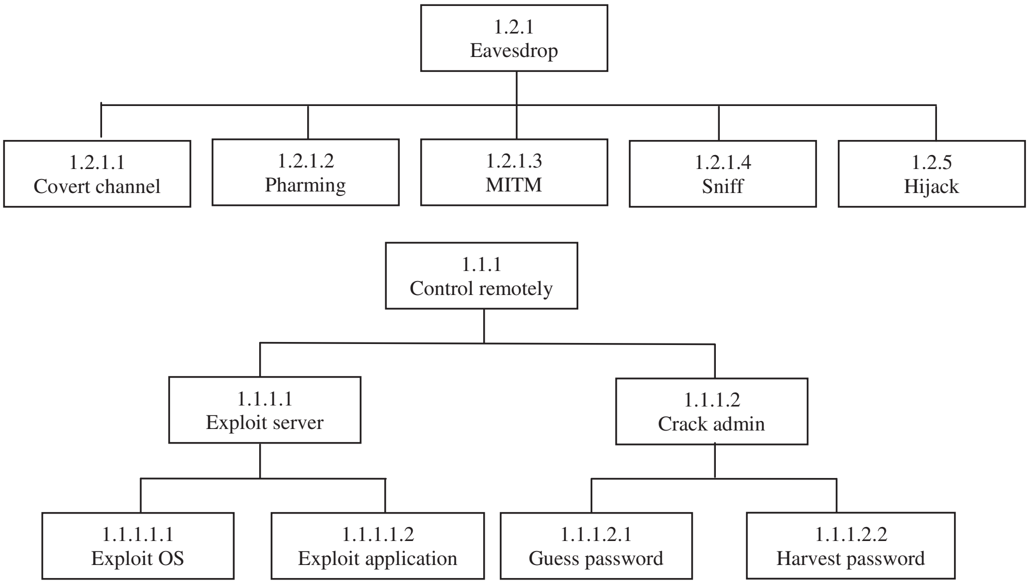


Fig. 7. The comparison of the uncertainty of two attack subtrees: (a) the uncertainty of the "Control remotely" attacks is 1.53 and (b) the uncertainty of the "Eavesdrop on communication channels" attacks is 2.22.

following observations about the attack process can also be deduced from the quantitative analysis (check the information in Table 1).

- The remote attacks (e.g., "1.1.1.1.2 Exploit application vulnerabilities", "1.2.1.1 Setup covert channel") are much more complex than the physical and social engineering attacks (e.g., "1.2.3.1 Take advantage of phishing web sites", "1.1.2.3 Steal server hard drive").
- The workstations are more vulnerable to the attacks (e.g., "1.2.2 Take remotely control of terminals") than the server.
- The insiders are the sources of a small number of the attacks (e.g., "1.2.3.3 Bribe authorized user"). In fact, the vast majority of the attacks (e.g., "1.1.1 Take remotely control of server") are launched from external zone of the network by the outsiders.
- A large number of the attacks (e.g., "1.2.3.2 Send fake e-mails", "1.2.1.3 Employ man-in-the-middle technique") are due to the lack of the users' experience and competence. In fact, a smaller number of the attacks (e.g., "1.2.2.3.1 Circumvent access control mechanisms") are caused by the system vulnerabilities.

Such observations show the importance and usefulness of the quantitative metric. In view of these observations, the administrators of the network can take defensive countermeasures to make the network more robust against upcoming attacks. In this regard, there are a number of effective remedies for handling the security flaws of the network. For example, as stated, a large number of the attacks are due to the lack of the users' experience and competence. With education and caution, the human-made errors may highly be reduced, although they are always present, more or less, due to their human nature.

5. Conclusions

5.1. Summary

In this paper, we have utilized the concept of entropy and derived a quantitative model for the security analysis of computer systems. The evaluation process consists of two phases called static phase and dynamic phase. At first, the initial values of the uncertainty of all nodes of attack tree are individually evaluated; then, step by step, the values of the uncertainty of the nodes are propagated up to one higher level. The evaluation process continues until the root is reached. It implies that the uncertainty gradually increases towards the higher levels of the attack tree. The evaluation process continues until the root is reached. This model can be used to assess the amount of the uncertainty in the behavior of attackers. In other words, we want to have an indicator of how much the quantified amount of security will be close to absolute security. Due to the properties of entropy function, the defined metric has a probabilistic nature so that it can be made properly consistent with the context of security analysis. This type of analysis can be used to compare the security level of two possible versions of a system. In addition, it may help security administrators to identify more vulnerable components of their systems and then to take necessary defensive countermeasures. Finally, this work proposes a systematic method for an attack tree's parameterization so that it can be utilized for doing a variety of security analyses.

5.2. Further research

This work can be considered as first steps towards a general approach for the quantitative evaluation of systems from an information-theoretic perspective. Information theory has a rich and useful set of quantities. It seems that there is much more in the field of information theory that can be suitably developed and applied for handling different issues in the area of security quantification. For example, depending on a security analysis' purposes, it may be necessary to consider continuous aspects of the attacker behavior such as time durations of attack scenarios. In order to deal with such requirements, we can take the advantage of the concept of entropy of continuous random variables. Let X be a continuous random variable with the distribution function $F_X(x) = P(X \leqslant x)$ and the density function $f_X(x) = F'_X(x)$. The entropy $h(X)$ (also called differential entropy) of the random variable X is given by [33]:

$$h(X) = -\int_{-\infty}^{\infty} f_X(x) \log f_X(x) dx.$$

We may be able to extend this concept in such a way that it is applicable for evaluating security metrics such as "time durations of attack scenarios". Such metrics can give more insights into security issues of systems.

As another example, mutual information is also one of the most useful quantities in information theory. Consider two random variables X and Y. Mutual information quantifies the amount of dependence between the random variables X and Y and is defined to be:

$$I(X;Y) = H(X) - H(X|Y).$$

This notion gives us a good idea: If we are able to define one random variable for describing system behavior and one random variable for describing attacker behavior, then it would be possible to derive a quantitative metric of how much dependence there is between the attacker behavior and the system behavior.

Acknowledgment

We would like to thank the Research Institute for ICT of Iran for their financial support of this research.

References

[1] K. Sallhammar, B.E. Helvik, S.J. Knapskog, On stochastic modeling for integrated security and dependability evaluation, J. Netw. 1 (5) (2006) 31–42.

[2] B. Madan, K. Goseva-Popstojanova, K. Vaidyanathan, K.S. Trivedi, A method for modeling and quantifying the security attributes of intrusion tolerant systems, Perform. Eval. 56 (1–4) (2004) 167–186.

[3] K. Goseva-Popstojanova, F. Wang, R. Wang, F. Gong, K. Vaidyanathan, K.S. Trivedi, B. Muthusamy, Characterizing intrusion tolerant systems using a state transition model, in: Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX II), vol. 2, 2001, pp. 211–221.

[4] D. Wang, B. Madan, K.S. Trivedi, Security analysis of SITAR intrusion-tolerant system, in: Proceedings of the ACM Workshop on Survivable and Self-Regenerative Systems, 2003, pp. 23–32.

[5] F. Stevens, T. Courtney, S. Singh, A. Agbaria, J.F. Meyer, W.H. Sanders, P. Pal, Model-based validation of an intrusion-tolerant information system, in: Proceedings of the 23rd Symposium on Reliable Distributed Systems (SRDS'04), Florianpolis, Brazil, October 2004, pp. 184–194.

[6] S. Singh, M. Cukier, W. Sanders, Probabilistic validation of an intrusion-tolerant replication system, in: Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'03), June 22–25, 2003, pp. 615–624.

[7] M. Kaâniche, E. Alata, V. Nicomette, Y. Deswarte, M. Dacier, Empirical analysis and statistical modelling of attack processes based on honeypots, in: Proceedings of the Workshop on Empirical Evaluation of Dependability and Security (WEEDS'06), IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'06), Philadelphia, June 25–28, 2006, pp. 119–124.

[8] E. Jonsson, T. Olovsson, A quantitative model of the security intrusion process based on attacker behavior, IEEE Trans. Softw. Eng. 23 (4) (1997) 235–245.

[9] R. Ortalo, Y. Deswarte, M. Kaâniche, Experiments with quantitative evaluation tools for monitoring operational security, IEEE Trans. Softw. Eng. 25 (5) (1999) 635–650.

[10] M.A. McQueen, W.F. Boyer, M.A. Flynn, G.A. Beitel, Time-to-compromise model for cyber risk reduction estimation, in: Proceedings of the First Workshop on Quality of Protection, Quality of Protection: Security Measurements and Metrics, Springer, September 2005, pp. 49–64.

[11] D.J. Leversage, E. James, Estimating a system's mean time-to-compromise, IEEE Sec. Priv. 6 (1) (2008) 52–60. March 16–19.

[12] N. Paulauskas, E. Garsva, Attacker skill level distribution estimation in the system mean time-to-compromise, in: Proceedings of the First International Conference on Information Technology, Gdanks, May 18–21, 2008, pp. 1–4.

[13] M. Xu, S. Xu, An extended stochastic model for quantitative security analyses of networked systems, Inter. Math. 8 (3) (2012) 288–320.

[14] S. Basagiannis, P. Katsaros, A. Pombortsis, N. Alexiou, Probabilistic model checking for the quantification of DoS security threats, Comp. Sec. 28 (1) (2009) 450–465.

[15] C. Bodei, M. Curti, P. Degano, A quantitative study of two attacks, in: Proceedings of the 2nd International Workshop on Security Issues with Petri Nets and other Computational Models (WISP'04), Electronic Notes in Theoretical Computer Science, vol. 121, 2005, pp. 65–85.

[16] M.S. Ahmed, E. Al-Shaer, L. Khan, A novel quantitative approach for measuring network security, in: Proceedings of the 27th IEEE Conference on Computer Communications (INFOCOM'08), April 13–18, 2008, pp. 1957–1965.

[17] I. Cervesato, Towards a notion of quantitative security analysis, in: Proceedings of the First Workshop on Quality of Protection, 2005, pp. 12–26.

[18] A. Hecker, M. Riguidel, On the operational security assurance evaluation of networked IT systems, in: Proceeding of the 9th International Conference Smart Spaces and Next Generation Wired/Wireless Networking (NEW2AN'09), St Petersburg, Russia, September 2009.

[19] S. Jafari, F. Mtenzi, R. Fitzpatrick, B. O'Shea, Security metrics for e-healthcare information systems: a domain specific metrics approach, Int. J. Dig. Soc. (IJDS) 1 (4) (Dec. 2010) 238–245.

[20] L. Wang, T. Islam, T. Long, A. Singhal, S. Jajodia, An attack graph-based probabilistic security metric, in: Proceedings of the 22nd Annual IFIP WG 11.3 Working Conference on Data and Applications Security (DBSEC'08), Lecture Notes in Computer Science (LNCS), vol. 5094, 2008, pp. 283–296.

[21] W. Boyer, M. McQueen, Ideal based cyber security technical metrics for control systems, in: Proceedings of the 9th International Conference on Critical Information Infrastructures Security (CRITIS'07), Limassol, Cyprus, October 13–15, 2007, pp. 3–5.

[22] E. LeMay, M.D. Ford, K. Keefe, W.H. Sanders, C. Muehrcke, Model-based security metrics using Adversary view security evaluation (ADVISE), in: Proceedings of the 8th International Conference on Quantitative Evaluation of Systems (QEST'11), Aachen, Germany, September 5–8, 2011.

[23] R. Lippmann, J. Riordan, T. Yu, K. Watson, Continuous Security Metrics for Prevalent Network Threats: Introduction and First Four Metrics, Project Report IA-3, MIT Lincoln Laboratory, Lexington, MA, 22 May 2012.

[24] P.K. Manadhata, J.M. Wing, A formal model for a system's attack surface, in: S. Jajodia, A. Ghosh, V. Swarup, C. Wang, X.S. Wang (Eds.), Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats, Springer, 2011, pp. 1–28. Chapter 1.

[25] B.Z. Moayedi, M. Abdollahi Azgomi, A game theoretic framework for evaluation of the impacts of hackers diversity on security measures, Reliab. Eng. Syst. Safety 99 (1) (2012) 45–54.

[26] A. Jürgenson, J. Willemson, Computing exact outcomes of multi-parameter attack trees, Lect. Notes Comp. Sci. (LNCS) 5332 (2008) 1036–1051.

[27] C.K. Dimitriadis, Analyzing the security of Internet banking authentication mechanisms, Inform. Netw. Control J. 3 (2007) 34–41.

[28] S. Mauw, M. Oostdijk, Foundations of attack trees, in: International Conference on Information Security and Cryptology, Lecture Notes in Computer Science (LNCS), vol. 3935, 2005, pp. 186–198.

[29] J.H. Espedahlen, Attack trees describing security in distributed internet-enabled metrology, Master's Thesis, Department of Computer Science and Media Technology, Gjøvik University College, 2007.

[30] A. Buldas, T. Mägi, Practical security analysis of e-voting networks, in: Advances in Information and Computer Security, Second International Workshop on Security, Lecture Notes in Computer Science (LNCS), vol. 4752, 2007, pp. 320-335.

[31] K.S. Edge, A Framework for Analyzing and Mitigating The Vulnerabilities of Complex Networks Via Attack and Protection Trees, Ph.D. Thesis, Air Force Institute of Technology, Ohio, 2007.

[32] A. Jürgenson, J. Willemson, On fast and approximate attack tree computations, in: Information Security, Practice and Experience, Lecture Notes in Computer Science (LNCS), vol. 6047, 2010, pp. 56–66.

[33] T.M. Cover, J.A. Thomas, Elements of Information Theory, second ed., John Wiley & Sons, 2006.

[34] C. Diaz, S. Seys, J. Claessens, B. Preneel, Towards Measuring Anonymity, in: R. Dingledine, P. Syverson (Eds.), Designing Privacy Enhancing Technologies (PET'02), Lecture Notes in Computer Science (LNCS), vol. 2482, 2002, pp. 54–68.

[35] B. Hosp, P.L. Vora, An information-theoretic model of voting systems, Math. Comp. Model. 48 (9–10) (2008) 1628–1645.