RC24494 (W0802-051) February 13, 2008
Computer Science

IBM Research Report

Risk Modulating Factors in Risk-Based Access Control for Information in a MANET

Pau-Chen Cheng, Paul A. Karger
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

Risk Modulating Factors in Risk–Based Access Control for Information in a MANET*

Pau–Chen Cheng
pau@us.ibm.com

Paul A. Karger
karger@watson.ibm.com

IBM Thomas J. Watson Research Center

March 3, 2008

Abstract

We present an approach for evaluating risk using risk–contributing factors. This approach could be applied recursively to a hierarchy of risk–contributing factors. We also use a MANET scenario to demonstrate how the approach may be applied. We believe that the scenario covers some of the most important risk factors regarding access control for information in a MANET. The set of risk factors discussed in this paper is by no means complete. Besides the usual technical considerations for information system security, other factors such as human psychology, social network and warfare should also be taken into consideration to evaluate risk in a MANET and much more research is needed.

1 Introduction

Access control for information in a MANET (mobile ad–hoc network) is concerned about the risk of information leakage in a MANET environment. Intuitively, risk means that some unwanted events *may* happen in the *future* to cause damage or undesirable outcomes. Risk is viewed as a probabilistic notion since the exact future cannot really be known until it becomes the present and the past. Nonetheless, we human beings do attempt to predict the future from the past experience and the present

*This Research is conducted through participation in the International Technology Alliance sponsored by the U.S. Army Research Laboratory and U.K. Ministry of Defense and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the US Army Research Laboratory, the U.S. Government, the UK Ministry of Defense, or the UK Government. The US and UK Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

situation, examining multiple factors to plot and determine plausible courses and outcomes of future events. In this paper we try to do the same by decomposing a MANET into different components and examine the factors that may contribute to information leakage.

In reality, a MANET and its context are very dynamic, so are the factors contributing to risk. A MANET and its context would have a unique set of factors and each factor could have its unique impact on risk. The set and the impacts may all change with time. It would be very difficult to give a list all risk contributing factors and their impacts on risk. Rather, we present a basic approach for evaluating risk using risk contributing factors. A risk factor could be divided into factors of finer grain and risk factors of different levels of granularity could be arranged in a hierarchy, and the approach could be used recursively through the hierarchy. The eventual goal is to use the approach to build either an automated risk–based access control system or a decision support system where the decision maker could "walk" through the hierarchy to get different levels of details of risk analysis.

We use the approach to study a specific MANET scenario, in which a MANET node, the *sender* needs to make a risk–based access control decision on sending information to another node, the *receiver* through a *communication channel*. A node is decomposed into its information system, human user(s) and physical defense and a channel is decomposed into two end points; the risk associated with the end points, the information system, the users and the physical defense are examined.

The rest of this paper is organized as follows: section 2 presents the basics: the assumptions, the scenario, the risk model and the risk–evaluating approach. Section 3 briefly discusses related work. Section 4 examines risk factors in the communication channel, section 5 examines risk in the information system, section 6 examines risk factors associated with human users, section 7 examines risk associated with physical defense, and section 8 concludes.

2 The Basics

This section presents the basic assumptions, settings, risk model and methodology for risk evaluation discussed in this paper.

2.1 Basic Assumptions

While there has been considerable amount of research and practice with regard to information security, a MANET environment poses some unique challenges to protect information. These challenges become part of the basic assumptions of this paper outlined below:

- There is no fixed infrastructure to bootstrap any MANET–wide security measures. In particular, it is assumed there is no node trusted by all other nodes in a MANET. However, there could be strong mutual trust among a subset of the nodes; for example, strong trust among US and UK forces.
- There is no complete trust among nodes and personnel in general. However, there could be different degrees of trust of different aspects among nodes and personnel. For risk evaluation, a lower degree of trust generally leads to higher risk.

- Very little, if any physical security can be assumed. And it must be assumed that any node or personnel could be captured or compromised by adversaries unless there is strong enough evidence to indicate otherwise.
- Most, if not all, physical communication links are un–reliable and insecure wireless links.
- There could be a wide variance in nodes' capabilities to protect information. Some may have strongly temper–resistant hardware/firmware to establish secure communication channels and to protect information within the nodes, others may just have ordinary radios or cell phones. Two nodes communicating with each other can only settle for the greatest common denominator of their abilities to protect communication channels.
- A node has no real control over how other nodes handle and protect information. Therefore, when a node sends information to another node, it can at best hope that the receiving node will properly protect the information. *This implies that the sending node should evaluate the risk of information leakage from the receiving node before sending the information,* taking multiple risk–contributing factors at the receiving node into account, including the risk that the receiving node makes bad decisions regarding further distribution of the information. *The risk of information leakage from the communication channel must also be evaluated.*
- A node's need to serve human–beings could affect the effectiveness of its security measures. For example, a tamper–resistant node may protect its information using strong encryption. However, if the node needs to display the information in the clear to a human user, the strong protection offered by tamper–resistance and encryption may be rendered useless if the human user or the device with its display is captured by an adversary. On the other hand, if the node does not need to display its information, then tamper–resistance and encryption should be good enough to at least protect the confidentiality of the information.
- The adversaries have strong technical means to mount attacks. This does not imply that the adversaries are sophisticated technologists. With the ready availability of cheap but powerful portable computers, knowledge and tools, it is very possible that all an adversary needs is such a computer, the attack tools and the how–to instructions that can be freely downloaded from the Internet.
- Temporal considerations affect risk evaluation. For example, if it would take an adversary an hour to decipher an encrypted message but the message content will stay sensitive for only 20 minutes, then the risk of leakage through the channel is zero although the adversary will decipher the message with probability one.

2.2 Basic Setting

Figure 1 depicts the basic setting used in the discussion of MANET risk modulating factors in this article. The setting is that the sender would make risk–based decisions when providing information to the receiver through a communication channel. In particular, the sender is concerned about the damage that could result from leakage of sensitive information. Such leakage could happen through the communication channel, the information system or the human users on the receiver's side. The leakage could happen due to logical or physical compromise of the channel, the information system

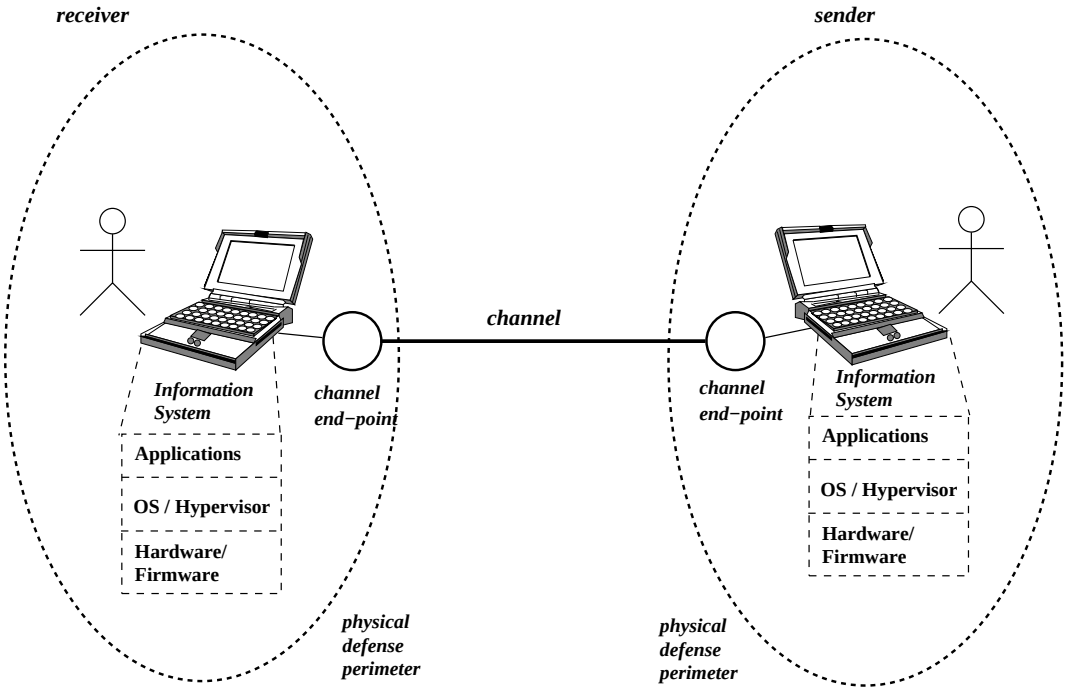

Figure 1: Basic Setting for MANET Information Security

or the human users. The malice or carelessness of a human user could also lead to leakage. From the sender's point of view, this report will examine the factors that may lead to information leakage in the channel, the receiver side information system, the human user(s) on the receiver side, and the effect of the strength of the physical security (or lack of it) on the receiver site.

The communication channel is modeled as two end points connected by a physical communication path consisting of one or more hops. The end points are responsible for establishing and maintaining a secure channel (on top of the path) between them, if they both are capable of doing so. The information system is modeled as a stack of hardware/firmware, hypervisor/operating system and applications. This stack may be tamper-resistant to some extent. The physical security is modeled by the physical defense perimeter. We intentionally separate the end points of the communication channel from the information systems. In practice, this separation may be either logical or physical. It is not unusual to have special purpose hardware/firmware to provide communication and communication security functions to an information system; alternatively these functions could be mostly implemented by software modules and applications on the information systems. This separation makes it easier to treat the risk factors in the channel as independent of the risk factors in the information systems.

Of course, the receiver would also be concerned about the genuineness of the received information. The concerns include the identity of the sender and if the information has been leaked or modified in transition. However, we believe that receiver would be concerned of the same kind of factors that are of concern to the sender.

2.3 Basic Risk Model

This report uses the usual definition of risk as the *expected value of damage* [1].

$$risk \equiv (P = probability\ of\ incurring\ damage) \times (value\ of\ damage) \quad (1)$$

The value of damage would be very dependent on the context of the mission. But the probability would be much more dependent on the kind of settings described in section 2.2 and this report will focus on modeling this probability. We will follow the *principle of the weakest link*; i,e., the weakest component in the setting will determine the probability. We will decompose the setting in Figure 1 into a few components and examine each component separately, assuming each can be compromised independently of the others. In other words, the probability P in equation 1 will be treated as the joint probability computed from the following probabilities which are assumed to be independent of one another:

- P_{CH} : the probability that the communication is compromised.
- P_{IS} : the probability that the information system is compromised.
- P_{HU} : the probability that the human user is compromised.
- P_{PH} : the probability that the physical security is compromised.

It should be noted that P_{CH}, P_{IS} and P_{HU} exclude physical compromises that are covered by P_{PH}.

The assumption of independence may not be entirely accurate. For example, if an end point of the channel is an integral part of the information system, then compromise of the information system implies compromise of the channel. This would result in a slight overestimate of P; this should be fine from the point of view of security, especially given the fact that all these probabilities are estimates to begin with.

2.4 General Methodology

While the true probabilities cannot be accurately derived, it is feasible to estimate the probabilities using the following two–step process, if qualitative comparison between two accesses can be made to determine which access is more likely to result in leakage of the accessed information [1].

1. Encoding the comparison using a formula that computes *risk indices*, such that a larger index implies a higher likelihood of leakage. Since the range of indices is not constrained, they could offer a high *resolution* to encode the intuition behind the qualitative comparison. The indices can be put on a scale, and the scale can be *calibrated* such that some points on the scale correspond to real access scenarios. The calibrated scale provides a frame of reference for step 2.
2. Assigning probabilities of leakage to the indices in a way that is commensurate with experience, intuition and threat assessment. These probabilities should increase with the indices. These assignments should be fine tuned over time; but the indices can be kept fixed to make the fine tuning easier. Such probability assignments are guesses; but *all access control policies and decisions are guesses*, given the unpredictability of the future [1]. Also, research is being carried out

under Project 6[1] to learn risk–based policies from positive and negative decision examples[2]. The result of this research should be useful to the fine–tuning of the probability assignments.

Many factors contribute to risk and it may be difficult to design one index formula covering all factors. Such a formula will contain many tunable parameters and be difficult to maintain. We could first design the index and probability formulas for each factor and then divide these factors into smaller groups, such that the relationship among members of a group can be understood or at least conjectured. This will allow a group's joint probability to be computed, then treat the groups as independent and compute their joint probability. Conversely, such a group could be formed by dividing a factor into a group of sub–factors, and such division could be done recursively. Thus, groups of risk factors can be arranged in a hierarchy and the methodology presented in this section can be used *recursively* through the hierarchy.

With all the above said, we have to caution that in some cases it is not easy to come up with such indices due to either the limit of technology or the concern that it is much better to be conservative and use a binary, secure/insecure scale in certain cases. Also, it is possible a node may not be able to get reliable information about another node to compute risk indices, in which case one may have to rely on *trust* to compute risk indices, using the work done on *Trust Evaluation* which is part of ITA Project 6.

We would recommend the use of secure hardware technologies such as TCG [2, 3], secure co–processor [4, 5] and secure processor [6, 7] in MANET systems. Such technologies could enable a system to provide other systems in a MANET, with some level of confidence, the software and configuration that is currently running on it. Such information could be used in risk assessment.

3 Related Work

Britton and Brown presented a model for risk measurement within a RADAC (risk adaptive access control) [8] context in their master's thesis [9]. Their context and risk evaluation approach are different than those in this paper, but their set of risk factors are similar to those in our scenario.

Chivers and Clark presents a model for risk assessment in a distributed system [10], using *risk profiles* of components in the distributed system. Their model enables distributed, incremental risk assessment such that change in the risk profile in one component only requires re–evaluation of risk in components that are affected by the change. We did not use their model, but the examination of risk factors for a component in a MANET node is actually toward building the risk profile for that component. In this sense their work and our work are complimentary to each other.

[1] Trust and Risk Management in Dynamic Coalition Environments

[2] This research is conducted by Yow Tzu Lim from University of York, supervised by Professor John Clark. It started during Mr. Lim's 2007 ITA summer internship in IBM Thomas J. Watson Research Center, mentored by Dr. Pau-Chen Cheng.

4 P_{CH} : Risk Factors in the Communication Channels

As described in section 2.2, the two end points of communication are responsible for establishing and maintaining a secure channel on top of the physical communication path between them. The secure channel is to provide some degree of protection on the secrecy and integrity of messages exchanged between the end points. If any one end point is incapable of assuming such responsibility, then P_{CH} is 1 and there is no need to evaluate the other probabilities.

4.1 An Ideally Secure Channel

An ideally secure channel, which is not practically achievable, would have the following properties:

- Each end point knows and has 100% *assurance of the other end point's identity*.
- Messages sent through the channel have perfect *secrecy*. In other words, *only the two end points can see the content of the messages*.
- Messages sent through the channel have perfect *integrity*. In other words,
  - Any *modification* to a message in transit can be detected by its receiver.
  - Any message *inserted* into the channel by a third party can be identified by the receiver as not authentic.
  - Any *retransmission/replay* of a message can be detected by its receiver.
  - Any *reflection* of a message back to its sender can be detected by that sender.

  Although the main concern of this report is about risk of information leakage (secrecy), it has been shown that lack of integrity protection could lead to information leakage [11, 12, 13, 14].

An adversary may try to attack the underlying physical communication path to disrupt communication, but it cannot compromise the secrecy or integrity of messages transmitted through a perfectly secure channel.

4.2 Secure Channels in Real World

In reality, no channel is ideally secure and this imperfection introduces risk of information leakage. Since the physical communication links are assumed to be unreliable and insecure, cryptographic means have to be used to establish a secure channel. Communication through a secure channel happens through two phases [15] :

1. *Channel Establishment*: the two end points negotiate with each other using some pre–defined protocol to establish a *security association (SA)*, which is a set of information shared between the end points. The protocol is usually called a *key management protocol (KMP)* [16, 17, 18]. The information in an SA would include crypto algorithms to be used to protect message secrecy and integrity, *secret keys* to be used by the crypto algorithms, life time of the SA, authenticated identities of the end points, etc.

An SA is a realization of the secure channel. The secret keys should be shared exclusively between the two end points; compromise of these keys means compromise of any protection the secure channel may offer. A new SA should be negotiated periodically (*key refreshment*) to defeat cryptanalysis attacks and to limit the damage resulted from a compromised key.

2. *Data Communication*: the two end points send messages to one another, using the crypto algorithms and keys in the SA to protect the secrecy and integrity of the messages ; these algorithms are applied according to pre-defined protocols [19, 20, 21] which are selected during channel establishment.

4.3 Attacks on Secure Channels

To steal the information in messages transmitted through a secure channel, an adversary can usually mount the following kinds of attacks, in addition to attacks on the physical communication path which is assumed to be insecure.

- *Impersonation* : by convincing an end point that the adversary is the other end point.
- *Cryptanalysis* : by analyzing the messages transmitted during channel establishment and data communication to recover either the secret keys or the content of the messages, which are encrypted. It should be noted that it may be possible to recover the content of encrypted messages without knowing the secret keys; this is especially so when the channel does not offer adequate protection on message integrity.

To mount such attacks, an adversary can exploit:

- *protocol flaws* : the protocols used in channel establishment or data communication could have exploitable flaws [22]. For example, flaws in the protocol used in channel establishment could lead to impersonation, or to the selection of weak encryption algorithms or easily–compromised secret keys which are exploitable crypto flaws. Message content could be leaked if the security protocol used in data communication does not offer message integrity protection.
- *crypto flaws* : crypto algorithms/primitives used in KMP and data communication my have exploitable flaws.
- *implementation flaws* : The hardware/software/firmware implementing the KMP and secure data communication may have exploitable flaws. For example, a weak random number generator on one end point could lead to secret keys that are easily predicted or guessed.

Ideally, end points with such flaws should not be used. But in many cases, these flaws are not discovered until long after the protocols, crypto algorithms, or their implementations were deployed. A MANET may have to live with these flaws because updates or replacements may not be available.

4.4 Estimate P_{CH}

Based on the methodology outlined in section 2.4 and the discussion in section 4.3, we further decompose P_{CH} into three independent probabilities to account for the protocol flaws, crypto flaws and implementation flaws.

4.4.1 Protocol Flaws

Designing a key management protocol (KMP) used for secure channel establishment is tricky. It has been known for a long time that a KMP could be broken even if it uses only strong crypto primitives [23, 24]. For example, since there is usually more than one pair of end points executing a KMP, one pair may be used as an *oracle* to break another pair. The protocols for data communication have similar security concerns.

To evaluate the risk introduced by protocol flaws, we will list two sets of properties. The first set includes the "must have" properties for a protocol, and a protocol should be considered broken if it does not satisfy any property in the "must have" set. The second set includes the "should have" properties for a KMP; these properties may be desirable in certain situations. A protocol is rather than strong but not completely broken if it does not satisfy a property in the "should have' set.

The "must have" properties for a protocol are:

- secure against message insertions and modifications,
- secure against message replays,
- secure against message reflections,
- secure against impersonation.

Since the physical communication path is insecure, it is impossible to prevent insertions, modifications, replays or reflections from happening. But a secure protocol should detect and reject any such events; this could mean dropping a few messages, disrupted communication, or even not being able to establish a secure channel instead of establishing a compromised "secure" channel. A protocol which does not satisfy any of the "must have" properties should be assigned a very high risk index that would make P_{CH} close to 1.

The "should have" properties for a protocol are:

- *forward security* : the compromise of one crypto key used by a secure channel (during channel establishment or data communication) does not imply the compromise of any other crypto key that was, is or *will be* used by the channel. This means that these keys must be generated in such a way that an adversary could not find any correlation among them. More than one key will be used for a secure channel because there is a need to change the key on a regular basis, namely *key refreshment*, to defeat cryptanalysis or brute–force search.
- anonymity : there may be a need to protect the secrecy of the identities of the end points.

In general, a protocol which does not satisfy any of the "should have" properties should be assigned a medium to high risk index. However, the assignment is very context–dependent. If the secure channel is to exist for a short while, on the order of tens of minutes or less, forward security may not be required. In some cases anonymity is not required. In these cases the risk index assignment should be based on the "must have" properties.

4.4.2 Crypto Flaws

Modern cryptography [25] has made tremendous progress in the past 40 years, in terms of both providing useful crypto primitives to protect data and useful techniques

to attack the protection, a.k.a. *cryptanalysis*.

An instance of a particular kind of crypto primitive, be it block cipher, message authentication code (MAC), digital signature, public key encryption, or key exchange such as the Diffie-Hellman exchange [26] could be specified by its algorithm and a set of tunable parameters, such as key length, the size of the prime field, etc. Conceptually, for the instances of a kind of crypto primitive, there could be two bounds :

- the *maximum upper bound*, instances specified above this bound are considered to be *secure enough*; although there could be different degrees of strength. For example, AES [27, 28] using 256–bit keys should be stronger than AES using 128–bit keys.
- the *minimum lower bound*, instances specified below this bound are definitely broken. For example, block ciphers with key lengths of 56–bit or less are broken [29].

Both bounds will go upward with the advances in cryptanalysis and the raw computing power. While these two bounds do not always coincide, it is usually difficult to tell how secure are the instances lying in between due to the advances in cryptanalysis and the huge gain in raw computing power. Therefore, we would recommend that any instance of a crypto primitive specified below the primitive's maximum upper bound be assigned a large risk index that would make P_{CH} very close to 1. A medium or small risk index should be assigned to instances specified above their corresponding maximum upper bounds.

Since a secure channel would usually use more than one crypto primitive, its risk index due to crypto flaws should be the maximum of the risk indices of all the crypto primitives it uses.

4.4.3 Implementation Flaws

The implementation of a secure channel is basically the implementations of the two end points. To account for the risk introduced by the flaws in these implementations, we are basically trying to evaluate the quality of the implementations. Since the end points are IT systems as well, we would defer the discussion of this evaluation to section 5, where P_{IS} is discussed. However, we would discuss evaluating the implementation of the pseudo–random number generator here.

The most important and fundamental component for establishing a secure channel is a strong *pseudo–random number generator (PRNG)*. A PRNG on an end point is responsible to output strong pseudo–random numbers that will be used either directly as secret crypto keys or as the material to generate these keys. *The game is lost* if the PRNG is weak or compromised and its output could be easily recovered[3] .

Barak and Halevi gave analysis of a practical, provably secure model for constructing a robust and secure PRNG [30], whose basic construct is depicted in Figure 2. The PRNG in Figure 2 has two components:

- an *entropy extractor* that collects and extracts entropy from entropy sources and outputs a *random seed* which is a strong random number.
- a *stretcher* which adds the random seed into its internal state and "*stretches*" the entropy in its internal state to produce a stream of pseudo–random bits which

[3]For example, using brute–force search.

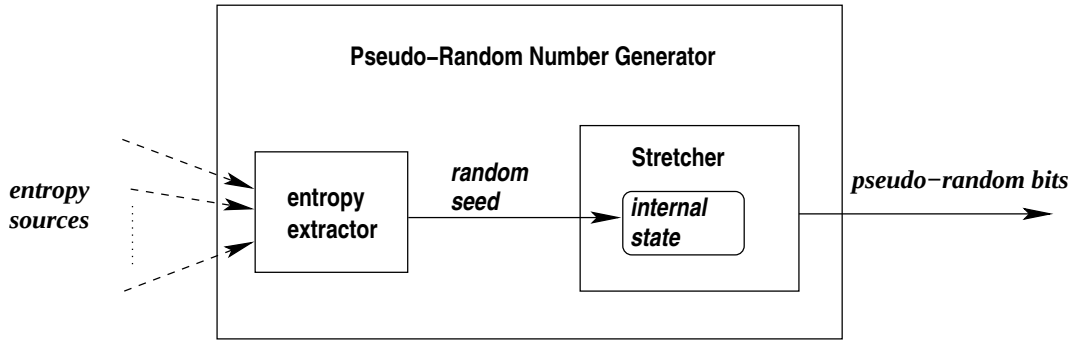

Figure 2: Basic Construct for a Secure and Robust PRNG

are *indistinguishable* from a stream of truly random bits for any entity with only limited[4] computing resources.

The entropy in the internal state is consumed as the output bits are produced; Barak and Halevi recommended performing a *refresh* operation every a few minutes to replenish the entropy in the internal state. This is done by adding a new random seed to the internal state.

The entropy extractor and the stretcher can all be implemented using readily available crypto primitives [30]. Gutmann also gave a good discussion on the system and implementation aspects of constructing PRNG's in chapter 6 of his book [31]. We must conclude that it is neither difficult nor expensive to build a secure and robust PRNG; nor would it be difficult or expensive to scrutinize [32, 33, 34] the PRNG to gain *high assurance* of it. The weakness of such a PRNG would most likely come from

- the weakness in the system upon which the PRNG is built. This weakness is discussed in section 5.
- the entropy sources. If these sources could not provide enough entropy or are controlled by the adversaries, the adversaries could easily predict the output of the PRNG. Entropy sources could come in three flavors:
  - A *dedicated, internal physical* entropy source such as instrumentation to sample the thermal noise in the voltage across a resistor or a reverse–biased diode.
  - Internal sources such as timings of asynchronous events or drifts in physical/mechanical parameters in the system upon which the PRNG is built. Examples are timings of interrupts, drifts in the internal clock, drifts in the rotation speed of hard drives, etc..
  - External sources such as the timings of arrivals of network packets and human key strokes, or drifts in physical environmental parameters such as temperature, humidity, etc..

  A dedicated, internal entropy source would be the preferred choice but most systems today do not have such dedicated entropy sources. Other sources may or may not provide enough entropy. In a hostile environment such like the ones in which a MANET is likely to operate, it is possible for the adversaries to exercise strong influence over, or even control these sources, especially the external sources.

[4] Polynomial time

Given these concerns, the *refresh* operation recommended in [30] may not be advisable in a MANET. Instead, we would recommend that a PRNG should be seeded with enough entropy prior to its deployment and that its *refresh* operation should be disabled upon deployment.

To assign a risk index to a PRNG, we would recommend considering the following properties of the PRNG, in descending order of importance:

1. It should have either a good internal entropy source or be seeded with enough entropy prior to its deployment.
2. Its design should follow proven principles, such as those outlined in [30] and [31].
3. Its design and implementation should be of high assurance gained through careful, formal scrutinization.
4. Its *refresh* operation should be disabled upon deployment in a potentially hostile environment unless it has an internal entropy source.

If any of the first 3 properties does not hold, the PRNG should be considered insecure and be assigned a high risk index; i.e., an index that would make P_{CH} close to 1. If the 4th property does not hold, the PRNG should be assigned a medium risk index.

To determine the risk index for the implementation of a secure channel, we recommend taking the maximum of the PRNG risk index and the risk index obtained by evaluating the over–all implementation of the end points.

4.4.4 Compute P_{CH} from Risk Indices

With the three risk indices from protocol, crypto and implementation flaws assigned, we would recommend computing P_{CH} in one of two ways :

1. assign a probability to each index and compute P_{CH} as the joint probability of the three probabilities by treating them as independent, or
2. choose the maximum of the three indices, assign a probability to this index and use this probability as P_{CH}.

5 P_{IS} : Risk Factors in Information Systems

Figure 3 depicts the *layered* structure of information systems; some information systems do not have the hypervisor layer. Risk in an information system (IS) comes from defects in these layers. These defects could be in the designs, implementations and configurations of these layers. Risk could also come from the environment in which the IS operates; the IS may be designed to operate correctly and securely under some assumptions that do not hold in the environment. Some of these assumptions may be *implicit*. For example, an application may be designed and implemented under the implicit assumption that the power supply will not fail and a power glitch may leave the application and its data in an inconsistent and insecure state.

Ideally, to build a secure IS, each layer should have well–defined interfaces to the layers above it and beneath it. And the internal of each layer should be of modular design and implementation, with each module having a well–defined interface and behavior. For the module to be secure, it should check all its input and reject any

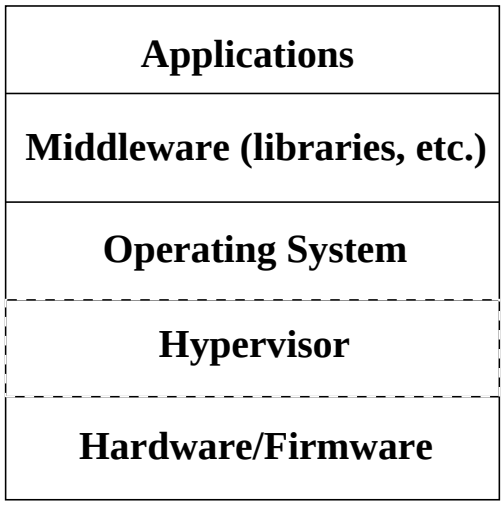

Figure 3: Layers of an Information System

input that it is not designed to handle, and always fails in a known state. In reality, very few IS's are designed in such a manner, let alone being implemented so; this coupled with the fact that each layer are usually very complex[5], means most IS's today are very insecure.

Many efforts have been expended on how to design, build and evaluate an IS so as to gain some level of assurance that the IS is secure. The first proposal for an evaluation criteria came in 1979 by Nibaldi [35]. Her proposal evolved into the the US Department of Defense *Trusted Computer System Evaluation Criteria* (*TCSEC*) [36]. In response to the US effort, the British [37], the Germans [38], and the French [39] all introduced evaluation criteria. In 1991, the UK, France, Germany, and the Netherlands introduced a European criteria [40], and in 1993, the Canadians followed suit [41]. For more details on the development of the evaluation criteria, see [42].

To stem the proliferation of mutually incompatible evaluation criteria, the *Common Criteria for Information Technology Security Evaluation* (*CC*) [43] was developed by a consortium of industrial and government partners and became an ISO standard. An evaluated IS is given a rating that indicates its level of security assurance. The highest rating for CC is *EAL7* and the lowest is *EAL1*. The highest rating for TCSEC is *A1* and the lowest is *D*. The low ratings mean no or very low assurance. The high ratings mean high or very high assurance; such a rating requires formal modeling and proof of the security properties of the IS; the process of designing and developing the IS must also be documented and reviewed. Thus, a high rating not only indicates a high level of security assurance, it is also an indication of the high quality of the design and development process. This high quality is necessary for high assurance because the evaluation process is not expected to find all security holes in an IS due to the complexity of the IS; but a well designed and developed IS is much less likely to have serious security holes.

It is generally desirable for an IS to have a security rating; but there are a few caveats :

[5]Considering the fact that modern OS's and applications are usually built with millions of lines of code.

- A rating does not necessarily imply a high degree of security. Some popular softwares, such as some versions of the Microsoft Windows™ and some versions of Linux, have gained the EAL4 rating. But we still see frequent announcements of security alerts and patches for these softwares. High assurance is generally viewed as at least EAL6 under the Common Criteria or at least B3 under the TCSEC. EAL5 and B2 are viewed as medium assurance under the Common Criteria and TCSEC, and ratings of EAL4 or B1 and below are viewed as low assurance.

  In practice, relative very few systems attained a high rating. Only three systems ever received a TCSEC A1 certificate, including the GEMSOS [44] secure OS, the SCOMP secure communications processor, and the Boeing MLS LAN. Under the Common Criteria, only some one-way network diode products from TENIX Defense Systems and from Compucat have received EAL7 certificates.

- A rating is only given to a specific configuration of a set of software/hardware that was evaluated. In other words, the rating does not cover the same set with a different configuration, nor does it cover a different set.

- Ratings (or Security) is *not composable*. A combination of two sets of software/hardware, each with a high rating, is not guaranteed to have a high rating. This combination, including its configuration, must be evaluated to gain a rating. Composite evaluation can be very difficult to accomplish as shown by Karger and Kurth [45].

The next question is how to decide how much to trust a system evaluated to a given level. Under the TCSEC, this was relatively easy, for two reasons. First, the requirements to achieve a given level certificate were the same for all systems. This meant that evaluations of different products could be compared to determine if system X had better or worse security than system Y. Second, the NSA published guidelines on how to decide what level of evaluated system was needed, given the highest level of classification of information stored on that system and the low level of security clearance of people allowed to use the system. These guidelines were published in the so-called Yellow Books [46, 47]. It is important to note that the Yellow Book recognized that even A1-evaluated systems are not perfect, and recommended that not even an A1 system was sufficient for some of the most risky configurations.

Comparing Common Criteria evaluations is much more difficult than comparing TCSEC evaluations for several reasons. First is that there is no counterpart of the Yellow Books for the Common Criteria. Second, a Common Criteria evaluation is much more complex than a TCSEC evaluation. There is no standard set of requirements for a given assurance level. Each product can define its own requirements as part of its *Security Target* document. Further, a given Security Target can optionally claim conformance to one or more *Protection Profiles*. A protection profile is intended to provide some means of comparison and standardized requirements, but since there can be many protection profiles, and each security target can specify additional requirements or can also waive certain requirements, the ability to compare two different Common Criteria evaluations is nearly non-existent.

With these caveats, we would recommend that a MANET node should choose its IS according to the following list of descending order of preference. The assignment of risk indices, quoted in "[ ]", is in ascending order.

- [very low] use an IS with a high security rating if the IS could fit the operational

need,

- [low to medium low] use a secure hypervisor (virtual machine monitor) with a high security rating to isolate different applications such that each application is placed in its own IS running in a separate hypervisor partition. While a secure hypervisor may not be able to prevent an application/IS from being compromised, it can prevent the compromise from spreading to other applications.

  Processor support for Hypervisor/Virtualization, such as Intel VT [48, 49], AMD–V [50] and IBM PowerPC virtualization [51, 52] is becoming increasingly common. Open–source hypervisors, such as XEN [53] and KVM [54] are also becoming popular. Active researches, such as sHype [55, 56] and SecVisor [57] are being conducted to make these hypervisors secure. The VAX VMM Security Kernel [58, 59] is a secure hypervisor that was targeted for the TCSEC A1 rating.

- [medium low to medium] use an IS with a medium or low medium security rating, such as TCSEC C2, B1 or EAL4 if the IS could fit the operational need,

- [medium to medium high] use an IS that has gone through and passed some systematic security scrutinization, preferably an IS that has been used in the field and found to be resilient against attacks.

- [very high] use an IS that could fit the operational need. Of course, no security assurance could be expected in this case.

As discussed in section 2.4, the sender may not be able to determine the properties of the receiver IS if the receiver cannot provide attestation to its IS using technology such as TCG/TPM [2, 3]. In which case the sender should either assume a high level of risk or assess the security/trustworthiness of the receiver using other means.

6 P_{HU} : Risk Factors in Human Beings

Traditionally, the risk of information leakage has been addressed by mandatory access control policies such as the MLS policies based on the Bell–Lapadula model [60]. Such a policy defines a static risk–benefit tradeoff and has a binary, allow/deny decision model; the policy is usually averse to risk by only allowing low–risk accesses. The new Fuzzy MLS model [1] proposed by the authors makes access control decisions based on quantified risk estimates of information leakage by human users; a decision could be one of *allow, deny or allow with a risk mitigation measure*.

To estimate P_{HU}, we would recommend using the rational of the Fuzzy MLS model and augmenting it for the MANET environment. Basically, risk of information leakage through a human user is incurred when the user accesses a piece of information. So P_{HU} is determined by comparing the profiles of the user and the information. For Fuzzy MLS, the profiles are the MLS labels of the user and the information. The Fuzzy MLS model compares such a pair of MLS labels to compute quantified estimates of two risk contributing factors, and uses these two factors to estimate P_{HU}. These two factors are :

- How *tempted* is the human user to leak the information? This is determined by comparing the user's level of trustworthiness (MLS clearance level) and the value of the information (MLS sensitivity level). The temptation and the likelihood of

leakage increases as the value of the information increases and decreases as the user's level of trustworthiness increases.

- How strong a *need* does the user have to access the information? In Fuzzy MLS, this is determined by comparing the category memberships of the user and the information. Fuzzy MLS allows fuzzy category memberships in the range [0, 1].

  Human beings are imperfect and even the most trustworthy users may leak information inadvertently. This kind of "slip of tongue" is always possible. If the user has a legitimate, strong need to access the information, this possibility should be accepted as the cost of conducting business or accomplishing a mission. If the user has only a marginal need or no need at all, then at least part of the possibility should be accounted for in P_{HU}.

In a MANET environment, more attributes in addition to the MLS labels should be taken into account to evaluate a user's levels of trustworthiness and his need to access. We assume that the profile of a piece of information already exists; this profile should tell the information's value/sensitivity, relevant topics, origins, restrictions, etc.. Eventually, this profile should be determined using the work on *Secure Information Flow* which is part of ITA Project 5[6].

The level of trustworthiness of a human user should eventually be determined using the work on *Trust Evaluation* which is part of ITA Project 6. Some initial results of this work are presented in [61].

We would recommend using three profiles to evaluate a human user's trustworthiness and need to access. The first profile $uPROF$ summarizes the characteristics of the user. The second profile $oPROF$ summarizes the characteristics of the user's long–term affiliation, namely the organization/military unit to which the user belongs, but may also be the user's social network, such as the user's family, tribe or village. The third profile $mPROF$ represents the characteristics of the user's short–term affiliation, namely the dynamic coalition/MANET to which the user currently belongs. The $uPROF$ will be used to produce a *baseline evaluation* of the user's levels of trustworthiness and need to access. The $oPROF$ and $mPROF$ are meant to address the inevitable possibility that the user's affiliations are likely to have access to the information received by the user; thus their main effect would be to decrease the baseline levels of trustworthiness and need to access and therefore increase P_{HU}. The final evaluation obtained by using all three profiles will be used with the profile of the information to estimate P_{HU}.

An $uPROF$ should include an user's attributes in the following *non–exclusive* list, subject to availability.

- attributes for evaluating the level of trustworthiness, in descending order of importance:
  - past records of leaking of information. Any such record would decrease the level of trustworthiness. The user should be considered totally un–trustworthy if the records demonstrate a habit of information leakage.
  - official MLS clearance level
  - assessments by others who have known and/or worked with the user.

[6] Efficient Security Architectures and Infrastructures

  - past experiences of using and protecting sensitive information. An inexperienced user is probably less competent in protecting information.
  - the rank and position within the user's organization or social network, assuming higher rank or position implies higher level of trustworthiness.
- attributes for evaluating the level of need to access, in descending order of importance:
  - role/responsibility within the user's current dynamic coalition.
  - job description within the user's long–term affiliation.
  - official MLS categories set

We would suggest using a weighted–sum approach to compute the levels of trustworthiness and need to access from the values of these attributes, but leave the exact weights for the experts to decide [9]. The value of an attribute may in term be derived from values of finer–grain sub–attributes.

An $oPROF$ represents the characteristics of the user's long–term affiliation, which may be a military unit, an NGO, a local tribe, etc.. This profile should contain attributes of this affiliation in the following *non–exclusive* list, subject to availability.

- attributes for evaluating the level of trustworthiness, in descending order of importance:
  - past records of leaking of information. Any such record would decreases the level of trustworthiness. The level of trustworthiness should be at most medium if the records demonstrate a habit of information leakage.
  - any practice in place to protect sensitive information. Lack of such practice would decreases the level of trustworthiness. The practice may be specified by a policy or could be just a "best practice".
  - the charter/mission of the organization. Is it in conflict with protecting sensitive information? For example, a news agency may be less inclined to protect sensitive information.
  - level of information system security (discussed in section 5),
  - level of physical security (discussed in section 7).
- attributes for evaluating the level of need to access:
  - the role/mission of this affiliation in the dynamic coalition. Does this role/mission imply the need to access the information? How strong is the need?
  - the charter/mission of the affiliation. Does the charter/mission imply the need to access the information? How strong is the need? We must differentiate the need to access from the desire to access. The need should be a judgment of the provider of the information, namely the sender, while the desire is from the affiliation.

Besides these attributes of the user's long term affiliation. We should also look at the following factors :

- how close is the relationship between the user and his/her long–term affiliation? The closer the relationship, the more weight the $oPROF$ has on determining the user's levels of trustworthiness and need to access.

- the life span of the sensitive information. Usually a shorter life span would mean the affiliation's profile carries less weight on determining the user's levels of trustworthiness and need to access.

The $mPROF$ of a MANET is the union of the $uPROF$es and $oPROF$es of the members of the MANET and the set of particular attributes of the MANET. This set should include attributes in the following *non–exclusive* list, subject to availability :

- attributes for evaluating the level of trustworthiness, in descending order of importance:
  - levels of information system security (discussed in section 5) of members of the MANET,
  - levels of physical security (discussed in section 7) of members of the MANET.
  - the mission of the MANET. Is it in conflict with protecting the sensitive information?
- attributes for evaluating the level of need to access:
  - the mission of the MANET. Does the mission imply the need to access the information? How strong is the need?

Besides these profiles, the *level of confidence in the user's identity* must also be taken into account. This level of confidence is established by the strength of the evidence and mechanisms that were used to authenticate the user's identity [62]. Lower level of confidence should further decrease the levels of trustworthiness and need to access determined by using the profiles.

It is important to note that this analysis of risk factors in human beings is not intended to identify any particular individual as a potential or actual risk. This is the realm of psychological analysis, background investigation, and counter-intelligence. Deciding whether a particular individual is or is not working for the "other side" is an extremely complex and often contradictory exercise [63, 64].

7 P_{PH} : Risk Factors in Physical Environment

Due to the existence of technologies of building tamper–resistant hardware [4, 5] and secure processors [6, 7] for information systems, the evaluation of P_{PH} could be separated into the evaluation the tamper–resistance the IS and the evaluation of the strength of the physical defense of the MANET node in which the IS resides.

For a tamper–resistant IS to protect information, it should have the following properties:

- its software stack should be evaluated to have a high level of security assurance. This is discussed in section 5.
- it should have a very strong ability to detect any attempt of physical tampering.
- a detected attempt of physical tampering should result in the erasure/destruction of data stored in the IS. The erasure/destruction could be accomplished by either erasing the data or erasing the crypto keys that are used to encrypt the data.

  Data erasure is very different from just deleting a file or overwriting a chunk of memory [65]. Their exist computer forensics technologies [66, 67, 68] to recover

"deleted" data. Open source and commercial forensics tools for such recovery are readily available [69, 70, 71, 72]. Fortunately, technologies [73, 74] and tools [75, 76, 77] for erasing data also exist.

- there should be no effective *side channels*, such as power analysis [78] or EM radiation analysis [79] through which sensitive data on the IS can be recovered. More side channel technologies and counter measures have been published in recent years [80, 81, 82, 83].
- there should be no way for a human user to examine the data in the IS when the IS is deployed in a MANET. By "examine" we mean viewing, listening, or printing the data.
- there should be no physical or logical interface to download the data unless the data are strongly encrypted. The keys used to encrypt the data must not be downloadable.

  The only exception to this requirement is for administrating/maintaining/operating the IS, in which case a very strong authentication mechanism should be used to authenticate the administrator/maintainer/operator before the download is allowed to happen. Of course, these administrators/maintainers/operators should not be allowed to enter a dangerous environment.

If an IS has these tamper–resistant properties, then its P_{PH} should be low. Otherwise, its P_{PH} could be estimated by evaluating the likelihood of a breach of the MANET node's physical defense. Due to our lack of expertise in modern weaponry and warfare, we hesitate to formulate a model or an approach to evaluate the strength of the physical defense of a MANET node. We would make the following observations:

- It is possible to do at least a coarse rating on the strength of physical defense of a MANET node. For example, the rating for a modern main battle tank or a well-fortified position could be high, the rating for a light armored vehicle could be medium, and the rating for an infantry man with a rifle is low.
- The actual likelihood of a breach is a function of the strength of physical defense and the ability of the adversary. As demonstrated by the asymmetric, unconventional warfare conducted by the insurgency in Iraq, an adversary needs the tool and tactic to penetrate a weak point in the defense. No defense is without weakness and the adversary does not have to be the equal of the US/UK armed forces in terms of technology or firepower. And both sides in a conflict will learn and adopt. Thus the relationship between the adversary and the defender would be a very complex and evolving one.

  It may be possible to model the adversary as a subject to access an object, the defense. But the models of the subjects and the objects, and the modes of access would be much more complex than those in an MLS access control system. Much more research is needed with major input and effort from military experts.

8 Conclusion

This paper presents an approach for evaluating risk using risk–contributing factors. This approach could be applied recursively to a hierarchy of risk–contributing factors. We also use a MANET scenario to demonstrate how the approach may be applied. We

believe that the scenario covers some of the most important risk factors regarding access control for information in a MANET. The set of risk factors discussed in this paper is by no means complete. Besides the usual technical considerations for security and risk in an information system, other factors such as human behavior, psychology, social network and warfare must be taken into consideration to evaluate risk in a MANET. Much more research and validation are needed for this approach to make it truly useful for a MANET. Especially, help and input from military experts are needed to evaluate the risk resulting from the breach of physical defense.

9 Acknowledgment

We would like to thank Shai Halevi, Hugo Krawczyk, Pankaj Rohatgi and Dakshi Agrawal for many useful advises, reviews and discussions. We would also like to thank Howard Chivers and John A. Clark for providing their paper on risk assessment to us.

References

[1] Pau-Chen Cheng, Pankaj Rohatgi, Claudia Keser, Paul A. Karger, Grant M. Wagner, and Angela Schuett Reninger. Fuzzy Multi–Level Security: An Experiment on Quantified Risk–Adaptive Access Control. In *IEEE Symposium on Security and Privacy*, May 2007. (See also [84]). 5, 15

[2] Reiner Sailer, Xiaolan Zhang, Trent Jaeger, and Leendert van Doorn. Design and Implementation of a TCG–based Integrity Measurement Architecture. In *13th Usenix Security Symposium*, August 2004. http://www.usenix.org/events/sec04/tech/sailer.html. 6, 15

[3] Trusted Computing Group. http://www.trustedcomputinggroup.org. 6, 15

[4] Joan Dyer, Mark Lindemann, Ronald Perez, Reiner Sailer, Leendert van Doorn, Sean Smith, and Steve Weingart. Building the IBM 4758 Secure Cryptographic Coprocessor. *IEEE Computer*, pages 57–66, October 2001. http://www.cs.dartmouth.edu/~sws/pubs/comp01.pdf. 6, 18

[5] Steve Weingart. The IBM 4758 Secure Cryptographic Coprocessor Hardware Architecture and Physical Security, 1999. http://www.cl.cam.ac.uk/research/security/seminars/1999/materials/weingart-19990222b.pdf. 6, 18

[6] IBM building security into chips. http://hardware.silicon.com/pdas/0,39024643,39157961,00.htm. 6, 18

[7] Divya Arora, Srivaths Ravi, Anand Raghunathan, and Niraj K. Jha. Architectural Enhancements for Secure Embedded Processing. In *Proceedings of NATO Workshop on Security and Embedded Systems*, August 2005. http://palms.ee.princeton.edu/PALMSopen/arora05architectural.pdf. 6, 18

[8] Rahim Choudhary. A Policy Based Architecture for NSA RAdAC Model. In *Proceedings of the IEEE Workshop on Information Assurance and Security*, 2005.

http://cc1.sctc.mnscu.edu/infosec/WestPointWorkshop2005/cdrom/PDFs/Papers/S13P03.pdf, see also [99, 100]. 6

[9] David W. Britton and Ian A. Brown. A Security Risk Measurement for the RAdAC Model. Master's Thesis, US Naval Postgraduate School, Monterey, California, USA, March 2007. http://handle.dtic.mil/100.2/ADA467180. 6, 17

[10] Howard Chivers and John A. Clark. Risk Profiles and Distributed Risk Assessment. private communication, 2007. 6

[11] A Key Recovery Attack on the 802.11b Wired Equivalent Privacy Protocol (WEP). *ACM Transactions on Information and System Security (TISSEC)*, 7(2):319–332, May 2004. http://delivery.acm.org/10.1145/1000000/996948/p319-stubblefield.pdf. 7

[12] Nikita Borisov, Ian Goldberg, and David Wagner. Intercepting Mobile Communications: The Insecurity of 802.11. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MOBICOM)*, pages 180–189, 2001. http://www.isaac.cs.berkeley.edu/isaac/mobicom.pdf. 7

[13] Hugo Krawczyk. The order of encryption and authentication for protecting communications (Or: how secure is SSL?). In *Advances in Cryptology – CRYPTO 2001, Lecture Notes in Computer Science, Vol. 2139*, pages 216–233. Springer–Verlag, J. Kilian, 2001. see also [85]. 7

[14] Steven M. Bellovin. Problem areas for the IP security protocols. In *the 6th USENIX Security Symposium*, July 1996. https://www.usenix.org/publications/library/proceedings/sec96/full_papers/bellovin/bellovin.ps. 7

[15] S. Kent and K. Seo. Security Architecture for the Internet Protocol. IETF RFC 4301, December 2005. http://www.ietf.org/rfc/rfc4301.txt. 7

[16] C. Kaufman (editor). Internet Key Exchange (IKEv2) Protocol. IETF RFC 4306, December 2005. http://www.ietf.org/rfc/rfc4306.txt. 7

[17] Dan Harkins and Dave Carrel. The Internet Key Exchange (IKE). IETF RFC 2409, November 1998. http://www.ietf.org/rfc/rfc2409.txt. 7

[18] Pau-Chen Cheng. An architecture for the Internet Key Exchange Protocol. *IBM System Journal*, 40(3), 2001. http://www.research.ibm.com/journal/sj/403/cheng.html. 7

[19] S. Kent. IP Encapsulating Security Payload (ESP). IETF RFC 4303, December 2005. http://www.ietf.org/rfc/rfc4303.txt. 8

[20] Stephen Kent and Randall Atkinson. IP Encapsulating Security Payload (ESP). IETF RFC 2406, November 1998. http://www.ietf.org/rfc/rfc2406.txt. 8

[21] Pau-Chen Cheng, Juan A. Garay, Amir Herzberg, and Hugo Krawczyk. A security architecture for the Internet Protocol. *IBM Systems Journal*, 37(1), 1998. http://www.research.ibm.com/journal/sj/371/cheng.html. 8

[22] David Wagner and Bruce Schneier. Analysis of the SSL 3.0 protocol. In *Proceedings the 2nd USENIX Workshop on Electronic Commerce*, 1996. http://www.cs.berkeley.edu/~daw/papers/ssl3.0.ps. 8

[23] Ray Bird, Inder Gopal, Amir Herzberg, Phil Jason, Shay Kutten, Refik Molva, and Moti Yung. The KryptoKnight Family of Light–Weight Protocols for Authentication and Key Distribution. *IEEE/ACM Transc. on Networking*, 3(1):31–41, February 1995. 9

[24] Ray Bird, Inder Gopal, Amir Herzberg, Philippe A. Jason, Shay Kutten, Refik Molva, and Moti Yung. Systematic Design of a Family of Attack–Resistant Authentication Protocols. *IEEE Journal on Selected Areas in Communications*, 11(5):679–693, June 1993. 9

[25] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *HANDBOOK of Applied Cryptography*, pages 41 – 44. CRC Press, New York City, New York, USA, 1996. 9

[26] Whitfield Diffie and Martin E. Hellman. New Directions in Cryptography. *IEEE Transc. on Information Theory*, IT–22(6):644–654, November 1976. 10

[27] Federal Information Processing Standards Publication 197 (FIPS–197). *Specification for the ADVANCED ENCRYPTION STANDARD (AES)*. National Institute of Standards and Technology (NIST), 2001. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf. 10

[28] ADVANCED ENCRYPTION STANDARD (AES). http://csrc.nist.gov/archive/aes/index.html. 10

[29] Electronic Freedom Foundation. *Cracking DES: Secrets of Encryption Research, Wiretap Politics & Chip Design*. O'Reilly Associates, Sebastopol, CA, USA, 1998. 10

[30] Boaz Barak and Shai Halevi. An architecture for robust pseudo-random generation and applications to /dev/random. In *Proceedings of the 12th ACM conference on Computer and communications security*, 2005. http://doi.acm.org/10.1145/1102120.1102148 (See also [89]). 10, 11, 12

[31] Peter Gutmann. Random Number Generation. In *Cryptographic Security Architecture : Design and Verification*, chapter 6. Springer–Verlag, 2002. 11, 12

[32] Zvi Gutterman, Benny Pinkas, and Tzachy Reinman. Analysis of the linux random number generator. In *Proceedings of the 2006 IEEE Symposium on Security and Privacy*, 2006. see also [86]. 11

[33] Leo Dorrendorf and Benny Pinkas. Cryptanalysis of the Windows Random Number Generator. In *Proceedings of the 14th ACM Conference on Computer and Communication Security (CCS 2007)*, 2007. see also [87]. 11

[34] W. Schindler. *Functionality classes and evaluation methodology for deterministic random number generators, AIS 20, Version 1.* Bundesamt für Sicherheit in der Informationstechnik (BSI), Bonn, Germany, December 1999. http://www.bsi.bund.de/zertifiz/zert/interpr/ais20e.pdf. 11

[35] G. H. Nibaldi. Proposed Technical Evaluation Criteria for Trusted Computer Systems. Technical Report M79-225, The MITRE Corporation, Bedford, MA, 25 October 1979. http://csrc.nist.gov/publications/history/niba79.pdf. 13

[36] US Department of Defense Trusted Computer System Evaluation Criteria, DOD 5200.28–STD, December 1985. http://csrc.nist.gov/publications/history/dod85.pdf. 13

[37] UK Systems Security Confidence Levels. CESG Memo No. 3, Communications-Electronics Security Group, Cheltenham, UK, February 1989. 13

[38] Criteria for the Evaluation of Trustworthiness of Information Technology (IT) Systems. Technical report, Zentralstelle für Sicherheit in der Informationstechnik, Bonn, Germany, January 1989. 13

[39] Critères Destinés à Évaluer le Degré de Confiance des Systèmes d'Information. Technical Report 692/SGDN/DISSI/SCSSI, Service Central de la Sécurité des Systèmes d'Information, France, July 1989. 13

[40] Information Technology Security Evaluation Criteria (ITSEC). Version 1.2, Commission of the European Communities, Brussels, Belgium, June 1991. http://www.ssi.gouv.fr/site_documents/ITSEC/ITSEC-uk.pdf. 13

[41] The Canadian Trusted Computer Product Evaluation Criteria. Version 3.0e, Canadian System Security Centre, Communications Security Establishment, Ottawa, ON, Canada, January 1993. 13

[42] Marvin Schaefer. If A1 is the Answer, What was the Question?: An Edgy Naf's Retrospective on Promulgating the Trusted Computer Systems Evaluation Criteria. In *2004 Annual Computer Security Applications Conference*, pages 204–228, Tucson, AZ, 6-10 December 2004. IEEE. http://www.acsac.org/2004/papers/ClassicPaperSchafer.pdf. 13

[43] Common Criteria for Information Technology Security Evaluation. http://www.commoncriteriaportal.org/, see also [88]. 13

[44] National Computer Security Center, Ft. George G. Meade, Maryland, USA. *Final Evaluation Report for the Gemini Trusted Network Processor*, 28 June 1995. Report No. 34–94, http://www.aesec.com/eval/NCSC-FER-94-008.pdf, see also [90, 91]. 14

[45] Paul A. Karger and Helmut Kurth. Increased Information Flow Needs for High-Assurance Composite Evaluations. In *Second IEEE International Information Assurance Workshop*, pages 129–140, Charlotte, NC, 8-9 April 2004. 14

[46] Computer Security Requirements – Guidance for Applying the Department of Defense Trusted Computer System Evaluation Criteria in Specific Environments. Technical Report CSC-STD-003-085, DoD Computer Security Center, Ft. George G. Meade, MD, 25 June 1985. http://www.radium.ncsc.mil/tpep/library/rainbow/index.html. 14

[47] Technical Rationale Behind CSC-STD-003-85: Computer Security Requirements – Guidance for Applying the Department of Defense Trusted Computer System Evaluation Criteria in Specific Environments. Technical Report CSC-STD-004-085, DoD Computer Security Center, Ft. George G. Meade, MD, 25 June 1985. http://www.radium.ncsc.mil/tpep/library/rainbow/index.html. 14

[48] G. Neiger, A. Santoni, F. Leung, D. Rodgers, and R. Uhlig. Intel Virtualization Technology: Hardware Support for Efficient Processor Virtualization. *Intel Technology Journal*, 10(3), August 2006. http://www.intel.com/technology/itj/2006/v10i3/1-hardware/1-abstract.htm, see also [92, 93]. 15

[49] Vineet Chadha, Ramesh Illiikkal, Ravi Iyer, Jaideep Moses, Donald Newell, and Renato J. Figueiredo. I/O processing in a virtualized platform: a simulation-driven approach. In *Proceedings of the 3rd international conference on Virtual execution environments*, 2007. http://delivery.acm.org/10.1145/1260000/1254827/p116-chadha.pdf. 15

[50] AMD Virtualization Technology. http://www.amd.com/us--en/Processors/ProductInformation/0,,30_118_8796_14287,00.html, see also [94]. 15

[51] W. J. Armstrong, R. L. Arndt, D. C. Boutcher, R. G. Kovacs, D. Larson, K. A. Lucke, N. Nayar, and R. C. Swanberg. Advanced virtualization capabilities of POWER5 systems. *IBM Journal of Research and Development*, 49(4/5), July/September 2005. http://www.research.ibm.com/journal/rd/494/armstrong.pdf, see also [95, 96, 97]. 15

[52] Dave Boutcher and Dave Engebretsen. Linux Virtualization on IBM POWER5 Systems. In *Linux Symposium*, 2004. http://www.linuxsymposium.org/proceedings/reprints/Reprint-Boutcher-OLS2004.pdf. 15

[53] Xen Org. http://www.xen.org. 15

[54] KVM Wiki. http://kvm.qumranet.com/kvmwiki. 15

[55] Reiner Sailer, Trent R, Enriquillo Valdez Jaeger, Ronald Perez, Stefan Berger, John L. Griffin, Leendert P. Van Doorn, and Ramon Caceres. Building a MAC-based Security Architecture for the Xen Opensource Hypervisor. In *21st Annual Computer Security Applications Conference (ACSAC)*, September 2005. http://www.acsa-admin.org/2005/papers/171.pdf. 15

[56] John L. Griffin and Stefan Berger and Kenneth A. Goldman and Trent R. Jaeger and Ronald Perez and David R. Safford and Reiner Sailer and Enriquillo Valdez and Leendert P. Van Doorn and Xiaolan Zhang. Secure Foundations for Mission-Critical Computing. CIIP – 2nd Japan/U.S. Workshop on Critical Information Infrastructure Protection, May 2005. http://www.cs.stevens.edu/~rwright/JapanUS/Talks/griffin.pdf. 15

[57] Arvind Seshadri, Mark Luk, Ning Qu, and Adrian Perrig. SecVisor: A Tiny Hypervisor to Provide Lifetime Kernel Code Integirty for Commodity OSes. In *Proceedings of the 21st ACM Symposium on Operating Systems Principles (SOSP'07)*, pages 335–350, 2007. http://www.sosp2007.org/papers/sosp079-seshadri.pdf. 15

[58] P.A. Karger, M.E. Zurko, D.W. Bonin, A.H. Mason, and C.E. Kahn. A Retrospective on the VAX VMM Security Kernel. *IEEE Transactions on Software Engineering*, 17(11):1147–1165, 1991. see also [98]. 15

[59] P.A Karger. Multi–Level Security Requirements for Hypervisors. In *21st Annual Computer Security Applications Conference,* Tucson, AZ, pages 240–248. IEEE Computer Society, 2005. http://www.acsa-admin.org/2005/papers/154.pdf. 15

[60] David E. Bell and Leonard J. LaPadula. Computer Security Model: Unified Exposition and Multics Interpretation. Technical Report ESD–TR–75–306, The MITRE Corporation, Bedford, MA. HQ Electronic Systems Division, Hanscom AFB, MA, March 1976. http://csrc.nist.gov/publications/history/bell76.pdf. 15

[61] Dakshi Agrawal and Charanjit Jutla. Utility Sampling for Trust Metrics in PKI. International Association for Cryptologic Research (IACR) ePrint Archive, Report 2007/178, 2007. http://eprint.iacr.org/2007/178.ps. 16

[62] Johannes Helander and Benjamin Zorn. Medina: Combining Evidence to Build Trust. In *Workshop on Web 2.0 Security and Privacy 2007, Held in conjunction with the 2007 IEEE Symposium on Security and Privacy.* http://seclab.cs.rice.edu/w2sp/2007/papers/paper-183-z_2465.pdf and slides. 18

[63] Peter Wright and Paul Greengrass. *Spy Catcher: The Candid Autobiography of a Senior Intelligence Officer.* New York: Penguin Viking, 1987. 18

[64] Nigel West. *Mole-Hunt: The Full Story of the Soviet Spy in MI5.* London: Weidenfeld and Nicolson, 1987. 18

[65] Peter Gutmann. Data Remanence in Semiconductor Devices. In *Proceedings of the 10th USENIX Security Symposium*, 2001. http://www.usenix.org/events/sec01/full_papers/gutmann/gutmann.pdf. 18

[66] Computer Forensics World. http://www.computerforensicsworld.com/. 18

[67] Dan Farmer and Wietse Venema. Computer Forensics Class, August 1999. http://www.fish2.com/forensics/class.html. 18

[68] Jamie Morris. Forensics on the Windows Platform, 2003. http://www.securityfocus.com/infocus/1661 and http://www.securityfocus.com/infocus/1665. 18

[69] Dan Farmer and Wietse Venema. The Corner's Tool Kit. http://www.fish2.com/tct/. 19

[70] Derek Cheng. Freeware Forensics Tools for Unix, 2001. http://www.securityfocus.com/infocus/1503. 19

[71] Dan Farmer and Wietse Venema. Lazarus. http://www.fish2.com/forensics/lazarus.pdf. 19

[72] Open Source Forensics Tools for Windows. http://www.opensourceforensics.org/tools/windows.html. 19

[73] Peter Gutmann. Secure Deletion of Data from Magnetic and Solid-State Memory. In *Proceedings of the Sixth USENIX Security Symposium*, 1996. http://www.usenix.com/publications/library/proceedings/sec96/gutmann.html. 19

[74] National Industrial Security Program Operating Manual (NISPOM). US Depart of Defense. http://nsi.org/Library/Govt/Nispom.html. 19

[75] Eraser – Free secure data erase tool to wipe files on your hard drive. http://www.heidi.ie/eraser/. 19

[76] Destroying Data. Office of Information Technology, University of Minnesota. http://www1.umn.edu/oit/security/tools/OIT__12709_REGION1.html. 19

[77] How to securely erase the hard disk before selling ones computer, June 2006. http://linuxhelp.blogspot.com/2006/06/how-to-securely-erase-hard-disk-before.html. 19

[78] Suresh Chari, Charanjit S. Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards Sound Approaches to Counteract Power-Analysis Attacks. In *International Crytology Conference (CRYPTO)*, pages 398–412, 1999. http://link.springer.de/link/service/series/0558/bibs/1666/16660398.htm. 19

[79] Dakshi Agrawal, Bruce Archambeault, Josyula R. Rao, and Pankaj Rohatgi. The EM Side–Channel(s). In *4th International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, pages 29–45, 2002. http://link.springer.de/link/service/series/0558/bibs/2523/25230029.htm. 19

[80] Josyula R. Rao, Pankaj Rohatgi, Helmut Scherzer, and Stephane Tinguely. Partitioning Attacks: Or How to Rapidly Clone Some GSM Cards. In *IEEE Symposium on Security and Privacy*, 2002. 19

[81] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template Attacks. In *4th International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, pages 13–28, 2002. http://link.springer.de/link/service/series/0558/bibs/2523/25230013.htm. 19

[82] Dakshi Agrawal, Josyula R. Rao, and Pankaj Rohatgi. Multi–channel Attacks. In *5th International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, pages 2–16, 2003. http://www.springerlink.com/content/lywfabcb8w6c0d0r/. 19

[83] Dakshi Agrawal, Josyula R. Rao, Pankaj Rohatgi, and Kai Schramm. Templates as Master Keys. In *7th International Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, pages 15–29, 2005. http://dx.doi.org/10.1007/11545262_2. 19

[84] Pau-Chen Cheng, Pankaj Rohatgi, Claudia Keser, Paul A. Karger, Grant M. Wagner, and Angela Schuett Reninger. *Fuzzy Multi–Level Security: An Experiment on Quantified Risk–Adaptive Access Control*, 2007. IBM Research Report RC24190, http://domino.research.ibm.com/library/cyberdig.nsf/1e4115aea78b6e7c85256b360066f0d4/d2c93a2df2afd3968525728f00528d26?OpenDocument&Highlight=0,rc24190. 20

[85] Hugo Krawczyk. The order of encryption and authentication for protecting communications (Or: how secure is SSL?). International Association for Cryptologic Research (IACR) ePrint Archive, Report 2001/045, 2001. http://eprint.iacr.org/2001/045.ps. 21

[86] Zvi Gutterman, Benny Pinkas, and Tzachy Reinman. Analysis of the Linux Random Number Generator. International Association for Cryptologic Research (IACR) ePrint Archive, Report 2006/086, 2006. http://eprint.iacr.org/2006/086.pdf. 22

[87] Leo Dorrendorf, Zvi Gutterman, and Benny Pinkas. Cryptanalysis of the Random Number Generator of the Windows Operating System. International Association for Cryptologic Research (IACR) ePrint Archive, Report 2007/419, 2007. http://eprint.iacr.org/2007/419. 22

[88] CCMB–2006–09–001. *Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Models, version 3.1, revision 1*, September 2006. http://www.commoncriteriaportal.org/public/files/CCPART1V3.1R1.pdf. 23

[89] Boaz Barak and Shai Halevi. An architecture for robust pseudo-random generation and applications to /dev/random. International Association for Cryptologic Research (IACR) ePrint Archive, Report 2005/029, 2005. http://eprint.iacr.org/2005/029.pdf. 22

[90] R.R. Schell, T.F. Tao, and M. Heckman. Designing the GEMSOS Security Kernel for Security and Performance. In *the 8th National Computer Security Conference*, pages 108–119, Gaithersburg, Maryland, USA, 30 September – 3 October 1985. US DoD Computer Security Center and National Bureau of Standards. 23

[91] W.R. Shockley, T.F. Tao, and M.F. Thompson. An Overview of the GEMSOS Class A1 Technology and Application Experience. In *the 11th National Computer Security Conference*, pages 238–245, Baltimore, Maryland, USA, 17–20 October 1988. National Bureau of Standards/National Computer Security Center. 23

[92] Intel Virtualization Technology. http://www.intel.com/technology/itj/2006/v10i3/. 23

[93] *Intel 64 and IA–32 Architectures Software Developer's Manual, Vol. 3: System Programming Guide, Parts 1 and 2*, 2007.
Vol. 3A (Part 1): http://www.intel.com/design/processor/manuals/253668.pdf
Vol. 3B (Part 2): http://www.intel.com/design/processor/manuals/253669.pdf. 23

[94] *AMD64 Architecture Programmers Manual, Volume 2: System Programming, Publication No. 24593*, 2005. http://www.amd.com/us--en/assets/content_type/white_papers_and_tech_docs/24593.pdf. 24

[95] Dale Barrick, Ivan Berrios, Ron Carter, Ed Gerwill, Shashank Jamgavkar, Steve Mann, Andrei Matetic, Alain Plu, Ian Smith, and Nick Harris. *Logical Partitions on IBM PowerPC*. IBM Redbook SG24-8000-00, 2004. http://www.redbooks.ibm.com/redpieces/pdfs/sg248000.pdf. 24

[96] IBM. *PowerPC Microprocessor Family: The Programming Environments Manual for 64-bit Microprocessors, Version 3.0*, July 2005. http://www-01.ibm.com/chips/techlib/techlib.nsf/techdocs/F7E732FF811F783187256FDD004D3797/$file/pem_64bit_v3.0_2005jul15.pdf. 24

[97] Scott Vetter, Morten Vågmo, and Peter Wuestefeld M.A. *Advanced POWER Virtualization on IBM System p5: Introduction and Configuration*. IBM Redbook SG24-7940-02, 2004.

http://www.redbooks.ibm.com/redbooks/SG247940/wwhelp/wwhimpl/js/html/wwhelp.htm. 24

[98] P.A. Karger, M.E. Zurko, D.W. Bonin, A.H. Mason, and C.E. Kahn. A VMM security kernel for the VAX architecture. In *Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy*, 1990. http://cs.ucla.edu/~kohler/class/06f-aos/ref/karger90vmm.pdf. 24

[99] Machon Gregory. A Mechanism for Risk AdaptiveAccess Control (RAdAC). National Information Assurance Research Laboratory (NIARL), March 2007. http://www.nsa.gov/selinux/papers/radac07.pdf. 21

[100] Ray Spencer, Stephen Smalley, Peter Loscocco, Mike Hibler, David Andersen, and Jay Lepreau. The Flask Security Architecture: System Support for Diverse Security Policies. In *Proceedings of the 8th USENIX Security Symposium*, 1999. http://www.usenix.org/events/sec99/full_papers/spencer/spencer.pdf. 21