Evaluating MAPSec by marking attack graphs

Kameswari Kotapati · Peng Liu · Thomas F. La Porta

© Springer Science+Business Media, LLC 2008

Abstract MAPSec has recently been introduced as a security protocol for mobile telecommunication networks in the midst of numerous threats and vulnerabilities. Our initial study reveals that MAPSec can only provide protection coverage to a minor portion of the total network vulnerabilities. Motivated by this discovery, we have devised a toolkit—Cellular Network Vulnerability Assessment Toolkit for Evaluation (eCAT) to identify: (1) Exact protection coverage of MAPSec, in terms of percentage of attacks prevented; (2) Other kinds of security protocols required in addition to MAPSec; and (3) The most vulnerable network areas. We use the results from eCAT in Coverage Measurement Formulas (CMF) to identify other vulnerabilities. Results from eCAT are dually useful in that they not only reveal MAPSec's limited effectiveness but also provide insights into overall network vulnerabilities.

Keywords Security protocol evaluation · MAPSec evaluation · eCAT · Attack graphs · Mobile telecommunication networks · Vulnerability assessment

1 Introduction

MAPSec (Mobile Application Part Security), a key new security protocol, was introduced to combat mobile telecommunication network vulnerabilities. Our initial study reveals that while the mobile telecommunication network suffers from numerous vulnerabilities [1], MAPSec is designed to only protect signaling messages. It follows that MAPSec can only protect a minor portion of total network vulnerabilities, and hence MAPSec alone is insufficient for protecting mobile telecommunication networks.

Accordingly, many questions arise regarding MAPSec's effectiveness, such as (1) What is the exact protection coverage of MAPSec, in terms of percentage of attacks prevented; (2) What are the other kinds of security mechanisms required to tackle the attacks that can evade MAPSec; and (3) Are there any network areas more vulnerable than others, and if so, does MAPSec protect them. Motivated by these questions, we have developed Cellular Network Vulnerability Assessment Toolkit for Evaluation (eCAT) to test the design principles of MAPSec.

eCAT uses graph technology, boolean probabilities, and Coverage Measurement Formulas (CMF) to: (1) Measure effective coverage; (2) Identify the types of required security mechanisms to protect the network; and (3) Identify the most vulnerable network areas (also called *hot-spots*).

eCAT uses *attack graphs* to capture the effective coverage of MAPSec. Although attack graphs for mobile telecommunication networks have been generated by approaches such as aCAT [2, 3], these approaches are limited in performing quantitative coverage analysis. In particular, they are defense agnostic and cannot measure the coverage of security protocols; and they cannot detect network areas most in need of protection. eCAT extends these approaches with a set of new capabilities. It uses a boolean probability

K. Kotapati (✉) · P. Liu · T. F. La Porta
The Pennsylvania State University, University Park,
PA 16802, USA
e-mail: kotapati@cse.psu.edu

P. Liu
e-mail: pliu@ist.psu.edu

T. F. La Porta
e-mail: tlp@cse.psu.edu

based *graph marking algorithm* to measure the effective coverage, and quantifies the same with CMF's.

Our graph based approach is motivated by a key observation that attack graphs can not only capture the effects of the potential attacks against a mobile telecommunication network, but can also capture the consequences of the security protocol in terms of its protection coverage. Such consequences may be used to evaluate the effectiveness of a security protocols, and more so, measure the protection coverage offered by the security protocol. Hence, attack graphs generated by eCAT provide a practical way to evaluate the effectiveness of security protocols. The merits of eCAT are as follows:

1. eCAT is *automatic* and *visual*, i.e., it automatically outputs (i) attacks enabled by corruption of data items (data poisoning) in messages, data sources, and service logic; (ii) protection coverage offered by the security protocol; and (iii) network hot-spots.
2. eCAT is *interactive* and allows users to indicate invalid pre-conditions necessary for attacks when the security protocol is applied, thereby allowing for evaluating the effectiveness of security protocols.
3. eCAT is *practical* as it provides attack graph results that may be directly used in our CMF formulas, to quantify protection coverage of security protocols.

Results obtained from eCAT's evaluation of MAPSec provide insights into the not only effectiveness of MAPSec but also the current state of mobile telecommunication network security. Please note that this is the first comprehensive analysis of potential effectiveness of MAPSec.

The rest of the article is organized as follows. In Sect. 2, we present the preliminaries, in Sect. 3, we present an overview of our approach, and in Sect. 4, we present our approach in detail. In Sect. 5, we present our Coverage Measurement Formulas, in Sect. 6, we present a detailed evaluation of our MAPSec, and in Sect. 7, we detail the findings from our evaluation. In Sect. 8 we present the related work and finally in Sect. 9 we conclude with a discussion.

2 Preliminaries

In this section, we present the preliminaries required to understand our rationale and our approach.

2.1 Mobile telecommunication network

Mobile telecommunication networks provide circuit switched and high speed packet data services for mobile devices. These networks provides numerous services to subscribers that are facilitated by network servers called *service nodes*. Service nodes are comprised of (1) a variety

of data sources (such as cached read-only, update-able, and shared data sources); and (2) service logic that performs functions, retrieves data items from data sources, and computes other data items. They include the Home Location Register (HLR), the Visitor Location Register (VLR), the Mobile Switching Center (MSC), and the Gateway Mobile Switching Center (GMSC).

These service nodes are distributed across a geographical region as illustrated in Fig. 2. Accordingly, every subscriber is assigned a geographical region called a *home network* from where they may roam to other *visiting networks*. The home network stores the profile and current location (pointer to VLR) of all subscribers assigned to it, in the HLR. In addition, each network administrative area is assigned a VLR. The VLR stores temporary data of subscribers currently roaming in its assigned area; this subscriber data is received from the HLR of the subscriber. Every VLR is associated with a MSC that acts as an interface between the radio system and the fixed network, and handles circuit switched services for subscribers currently roaming in its area. These geographically distributed service nodes function together by mutually exchanging signaling messages that contain data items.

Application of the MAP protocol: MAPSec is designed to securely transport a crucial application layer protocol called Mobile Application Part (MAP). The MAP protocol (implemented on the 'security free' Signaling System No. 7 family of protocols) is used by the various service nodes to exchange signaling messages. MAP is primarily used for message exchange involving subscriber location management, authentication, call handling, handovers (managing calls while moving), supplementary services (SS), short message (SM) services, etc. MAP messages are sent in the clear and may be used to carry a variety of data items some of which are confidential. To protect such confidential data items in MAP messages, MAPsec was introduced [4–7]. *The main thrust of MAPSec is to provide message data and origin authentication to MAP messages.* We defer a discussion of MAPSec to Sect. 3.

We illustrate the utility of MAP using the *call delivery service*. This basic service is used to deliver incoming calls to any subscriber with a mobile device regardless of their location. When a call is placed to a mobile subscriber, the call (signaling message IAM) is sent to the nearest GMSC which is in charge of routing calls and passing voice traffic between disparate networks (refer Fig. 1). Each signaling message contains data items used to invoke functions at the destination service nodes. For example, the IAM signaling message contains the 'called number' data item and is used to invoke the function that finds the assigned HLR (home network) of the called party at the GMSC. The GMSC uses the address of the HLR to inform it of the incoming call using the MAP signaling message SRI. The SRI MAP message contains data

Fig. 1 Signal flow for call delivery service

items such as the *called number* and the *alerting pattern*. The *alerting pattern* denotes the pattern (PACKET SWITCHED DATA, SHORT MESSAGE SERVICE OR CIRCUIT SWITCHED CALL) that is used to alert the called mobile subscriber.

As the HLR is aware of the location of the called subscriber, it requests call routing information (called roaming number) from the VLR that is in-charge of the area where the subscriber is currently roaming using the MAP message PRN. The HLR then downloads the incoming call profile of the subscriber (including *alerting pattern*) to the VLR. The VLR assigns a roaming number for routing the call and passes it on to the HLR (in MAP message PRN_ACK) which forwards it to the GMSC (in MAP message SRI_ACK). The GMSC uses this roaming number to route the incoming call (message IAM) to the MSC where the subscriber is currently roaming. This MSC requests the incoming call profile for the called subscriber (message SIFIC) from the VLR and receives the profile (including *alerting pattern*) in the Page MS message. The MSC uses the *alerting pattern* in the incoming call profile to determine the *page type* which is the manner in which to alert (message Page) the mobile station. Thus subscribers receive incoming calls irrespective of their locations in the network.

2.2 Threat model of eCAT

The threat model of eCAT encompasses any *active attack actions* such as buffer overflow and command injection that can cause data item corruption in (1) signaling messages; (2) caches; (3) database records; (4) local variables; or corruption of (5) service logic that computes data items or generates signaling messages containing data items. Readers interested in how these actions may be taken may refer to [1].

Our threat model considers ‘*system acceptable incorrect value corruption*’, a type of corruption in which corrupt values taken on system acceptable, albeit incorrect values. For example, consider the data item *alerting pattern* described previously. It may take values PACKET SWITCHED DATA, SHORT MESSAGE SERVICE, OR CIRCUIT SWITCHED CALL. If the call is circuit switched then the *alerting pattern* value must be CIRCUIT SWITCHED CALL. In system acceptable incorrect value corruption, the adversary may take any attack action that can switch the value of *alerting pattern* from CIRCUIT SWITCHED CALL to SHORT MESSAGE SERVICE, for example.

This type of system acceptable incorrect value corruption has many indirect and unforeseen effects due to the nature of telecommunication networks where data items are inter-related within and across service nodes services due to the exchange of signaling messages. Attacks that exhibit this effect are called *cascading attacks*. Cascading attacks are so named because local effects of corrupt data items propagate or cascade to data items on remote service nodes through vehicles such as signaling messages, cached data items, and shared databases.

Using Fig. 2, we illustrate such a cascading attack on the call delivery service using system acceptable incorrect value corruption of the data item *alerting pattern*. This attack is called *alerting attack*. Here, the adversary may use websites [8, 9] to refer to proprietorial working manuals of

Fig. 2 Alerting attack

service nodes such as the GMSC and insert the commands similar to the following. Such command insertion attacks have become highly common place, the most infamous attack being the telephone tapping of the greek government and top-ranking civil servants [10].

```
if(alerting_pattern .is. CIRCUIT SWITCHED CALL) then
    alerting_pattern = SHORT MESSAGE SERVICE;
end – if
```

The code presented attempts to switch the value of *alerting pattern* from the required value of CIRCUIT SWITCHED CALL to the SHORT MESSAGE SERVICE. Since the *alerting pattern* value is a system acceptable value, corruption not only remains undetected but also cascades. The corrupt *alerting pattern* is inserted into the MAP SRI message and is passed on to the HLR. The HLR, retrieves this corrupt *alerting pattern* value from the SRI message and passes it on to the VLR in the PRN MAP message. The VLR uses the corrupt *alerting pattern* to derive *page type*. For example, if the value of *alerting pattern* is SHORT MESSAGE SERVICE, then *page type* is assigned the value SMS.

The VLR outputs message Page MS to the MSC with corrupt data item *page type* which in-turn passes the corrupt *page type* to the base station which then alerts the subscriber. If *page type* is incompatible with the type of call (PACKET, CIRCUIT SWITCHED, etc.), the call session may not be received. This attack illustrates how corruption of a single data item at a single network location (such as the GMSC) results in propagation of corruption across the network (such as to the base station) through other corrupt data items, due to normal network operation.

Hence, a single action of changing the value of a data item has many diverse cascading effects across the network, and the eCAT threat model includes *any* attack actions that can cause system acceptable incorrect value corruption.

2.3 Using attack graphs to trace the effects of cascading attacks

In order to capture potential data corrupting attack actions and their potential effects in mobile telecommunication networks the Advanced Cellular Network Vulnerability Assessment Toolkit (aCAT) was developed [2]. As mentioned earlier, while aCAT can detect attack actions and their cascading effects presented in the form of attack graphs, aCAT cannot measure the coverage of security protocols. In this article, we extend aCAT to eCAT, which has security protocol evaluation capabilities. In this rest of this section, we will present a background on aCAT and attack graphs and in later sections, we describe how security protocol coverage is measured.

aCAT captures attack actions and effects by tracking data corruption within and across services nodes. For this purpose aCAT uses network knowledge from telecommunication specifications provided by the Third Generation Partnership Project [11], and is incorporated with a specially designed network dependency model and infection propagation rules. aCAT applies the inbuilt network dependency model and infection propagation rules on the network knowledge obtained from specifications to exhaustively identify all possible attack actions and their cascading effects. These attacks are presented in the form of easy to read attack graphs detailed next. For further information on aCAT, please refer to [2].

Attack graphs: We explain the terminology of attack graphs generated by aCAT, using a fragment of the attack graph (Fig. 3) produced for the above mentioned alerting attack.

The aCAT attack graph may be defined as an attack state transition starting with the pre-conditions, followed by attack actions and ending with its cascading effects. The aCAT attack graph use ‘nodes’ to represent network states,

Fig. 3 Fragment of unmarked attack graph for alerting attack generated by aCAT

and ‘edges’ to represent network transitions, with respect to the attack. Nodes may be either *pre-conditions* (occur at layer 0), or *attack effects* (occur at higher layers). Edges may be either due to adversary actions (connecting layer 0 and layer 1), or cascading effects (other layers). For description purposes, the attack graph has been divided into layers and assigned node labels.

Nodes at layer 0 correspond to the pre-conditions of the attack, such as *adversary’s physical access to network*, *adversary’s target*, and *adversary exploiting an existing vulnerability*. The adversary may have any of the following three levels of *physical access*: level 1: access to the air interface with the help of a physical device; level 2: access to links connecting central offices; and level 3: access to the service nodes. The adversary’s target is always a service node, and in-order for the attack to occur the adversary must exploit an existing vulnerability either in a data source, service logic or messages. In the attack graph for alerting attack shown in Fig. 3, the adversary could cause the alerting attack with level 3 physical access (Node 3), with target as the GMSC (Node 1), and exploit a vulnerability existing in the data source or service logic (Node 9).

aCAT attack graphs capture potential attacks by capturing all possible pre-conditions at layer 0 (and in turn their effects). For example we not only see the alerting attack due to corruption of data source or service logic at the GMSC, but also due to corruption of data source or service logic at the HLR (Nodes 3, 8, 9), VLR (Nodes 3, 4, 9, 10), MSC (Nodes 3, 6, 9, 12), and corruption of messages MAP SRI (Nodes 2, 8, 5, 11), MAP PRN (Nodes 4, 10, 5, 11), and Page MS (Nodes 5, 6, 11, 12). In capturing all potential conditions for the attack, aCAT can capture all potential attacks. Nodes at higher layers are effects of the attack propagating through the network. Effects typically include propagation of corruption between service nodes.

As each set of pre-condition nodes has different cascading effects, each set of pre-condition nodes and their effects are represented in the attack graph as an *attack tree*. Each attack tree may be thought of as an individual attack complete with pre-conditions, attack actions and effects. In the graph, trees are distinguished by the tree numbers assigned to its nodes. For example, all the nodes marked with number 1 belong to Tree 1 of the graph. The alerting attack graph has 7 attack trees, hence shows that the alerting attack can happen through 7 possible attacks. The graph in-turn is formed by merging multiple attack trees. The more trees a single attack action can cause, the more damage it can cause.

Attack graphs produced by aCAT are called ‘*unmarked attack graphs*’ since they are produced under the assumption that all attacks can happen and can cascade. *Unmarked attack graphs* do not indicate special conditions or areas such as hot-spots.

3 Overview of our attack graph marking approach

3.1 Overview of attack graph marking

The purpose of graph marking is to extract eliminated attacks from attack graphs, due to deployment of MAPSec. In marked attack graphs, shading is used to denote eliminated attacks, and hence unshaded nodes denote attacks evading MAPSec.

Such graph marking is possible by enabling the user to input the attack pre-conditions that MAPSec eliminates. Our graph marking algorithm then computes the attack effects that are eliminated by eliminating pre-conditions. Our graph marking scheme, detailed in Sect. 4, is efficient in computing these attack effects in large graphs. We can then use the CMF formulas to quantify protection coverage of MAPSec. We further detail the protection coverage quantification in Sect. 5. In the following we present MAPSec and its capabilities.

3.2 Threat model of MAPSEC

In this section, we present the threat model of MAPSec which is used to measure coverage of attacks.

Consider the MAP messages presented in the call delivery service. If an adversary gains access to cables connecting to the VLR, using readily available probes or protocol analyzers, they may capture the MAP message PRN (represented by Node 4) (which is sent in the clear) and modify the data item *alerting pattern* as mentioned previously and output it again with the corrupt *alerting pattern*. If MAP messages are authorized or encrypted, occurrence of such message based attacks may be eliminated.

MAPSec is used exactly for this purpose. It provides MAP message data authentication and origin authentication thereby preventing MAP message corruption and fabrication attacks. It provides a framework that allows ciphering, signing and authorization of MAP messages.

The MAPSec architecture is illustrated in Fig. 4. It requires a Key Administrative Center (KAC) to be maintained by every service provider. The function of the KAC is to negotiate MAPSec security associations on a network basis. The security association comprises of keys, algorithms, protection profiles and key lifetimes used to protect the MAP messages between the various service nodes. KAC’s use the Internet key exchange (IKE) protocol and Key Management Protocol (ISAKMP) framework to negotiate the security associations. The security associations are used to protect the MAP messages between the various service nodes by creating secure and ciphered MAP communication between the various service nodes.

Fig. 4 Architecture of MAPSEC

4 Graph marking: the algorithm

In this section, we present the rationale behind our approach, our graph marking algorithm, and CMF.

4.1 Graph marking rationale

The aim of the marking graphs is to (1) *extract* security protocol coverage from an unmarked attack graphs; (2) locate most vulnerable network areas or hot-spots; and (3) identify the other required security protocols in addition to MAPSec. This process is called *attack graph marking*.

In producing unmarked attack graphs we assume that all attack pre-conditions and their effects are possible. As deployment of some security protocols can eliminate some of the attack pre-conditions, we use *boolean probabilities* (1 and 0) to extract attack effects reduction. Using boolean probabilities enables eCAT to distinguish between existing (true or 1) and eliminated (false or 0) states (nodes in a attack graph) in an attack. By computing boolean probabilities for each node in the attack graph, eCAT can extract the attack effects that may be eliminated by deploying the security protocol, and estimate most vulnerable network areas.

4.2 Graph marking algorithm

In this section, we present our *graph marking algorithm* incorporated into eCAT (shown in Algorithm 1). We use our algorithm to illustrate how we extract MAPSec's coverage from an unmarked graph. Our algorithm is efficient in that it only requires as input (1) the unmarked attack graph; and (2) boolean probabilities for layer 0 nodes in the attack graph.

4.2.1 Input

Users of eCAT need only to set the boolean probability for layer 0 nodes based on the security protocols (e.g. MAPSec) under evaluation. Layer 0 nodes are the pre-conditions of the attack. Pre-conditions indicate the level of adversary's access to the network, vulnerability exploited by the adversary, and service node targeted. By indicating to eCAT if MAPSec can or cannot prevent the adversary from exploiting a certain vulnerability or gaining a certain level of network access, eCAT can compute if the attack effect can or cannot be prevented. This indication can be provided by setting the values of the pre-conditions to either 0 or 1. The number 0 indicates that the security protocols can prevent that pre-condition from existing.

For example, consider Node 5 in attack graph for the alerting attack shown in Fig. 5. This node occurs at layer 0 and is hence a pre-condition. This pre-condition states that the adversary can attack using level 2 access i.e. access to cables connecting service nodes. We know that MAPSec provides authentication of messages and source address checking, hence the usage of MAPSec eliminates the possibility of attack using level 2 access. Therefore, we set the boolean probability of this node to 0.

Accordingly, the number 1 indicates that the security protocols cannot prevent that pre-condition from existing. For example, consider Node 3 at layer 0 in the attack graph shown in Fig. 5. This pre-condition states that the adversary can attack using level 3 access i.e. direct access to service nodes. We know that MAPSec does not offer protection to data sources and service logic in the service node. Hence we set boolean probability of this node to 1.

4.2.2 Working

Our *graph marking* algorithm makes use of the boolean probabilities provided at layer 0 to compute probabilities and mark nodes at higher layers. This algorithm is particularly efficient when the unmarked attack graph is large as it is easy to visualize the coverage of the security protocol under evaluation using color coding invalid nodes. Our algorithm starts by computing probabilities for every node (indicated by line 1 in Algorithm 1) starting from layer 1 onwards up-to the maximum layer (indicated by line 2) in the attack graph.

Our graph marking algorithm starts by considering every node in layer 1. If the node belongs to a single tree (indicated by line 5) (e.g., Nodes 14, 15, 16) it means that this attack state exists due to corruption at a single network location. For example, Tree 2 (Node 14) shows attack effects due to corruption of a message 'MAP_SRI'. Hence

Algorithm 1 Graph marking

Input: G : Unmarked Attack Graph; P_0 : Set of probabilities for layer 0 nodes;
Output: $\bar{G}$: Marked Attack Graph i.e. attack graph with invalid node marking;
 Let N_G be the set of nodes in the unmarked attack graph G ;
 Let max_layer indicate the maximum number of layers in a attack graph;
Procedure Graph_Mark(G):

```

1: for ( $\forall \text{node}_i \in N_G$ ) do
2:   for ( $\ell \leftarrow 1$  to  $\text{max\_layer}$ ) do
3:     if  $\text{node}_i.\text{layer} == \ell$  then
4:       Set boolean result  $\leftarrow 1$ , temp  $\leftarrow 0$ ;
5:       if  $\text{node}_i.\text{tree\_no} == 1$  then
6:         {Node belongs to single tree}
7:         for ( $\forall p_j \in \text{Parent}(\text{node}_i)$ ) do
8:           result  $\leftarrow$  result  $\times p_j.\text{probability}$ ;
9:         end for
10:         $\text{node}_i.\text{probability} \leftarrow$  result;
11:       else
12:         {Node belongs to multiple trees}
13:         for ( $\forall t_k \in \text{node}_i.\text{tree\_no}$ ) do
14:           {Computing probability for individual tree}
15:           for ( $\forall p_j \in \text{Parent}(\text{node}_i)$ ) do
16:             if  $p_j.\text{tree\_no} == t_k$  then
17:               if  $p_j.\text{probability} == 0$  then
18:                 result  $\leftarrow 0$ ; break
19:               else
20:                 result  $\leftarrow 1$ ;
21:               end if
22:             {Computing combined probability for all trees}
23:             temp  $\leftarrow$  temp+result;
24:           end if
25:         end for
26:       end for
27:        $\text{node}_i.\text{probability} \leftarrow$  temp;
28:     end if
29:     {Marking Nodes}
30:     if  $\text{node}_i.\text{probability} == 0$  then
31:        $\text{node}_i.\text{mark} \leftarrow \text{invalid}$ ;
32:     else
33:        $\text{node}_i.\text{mark} \leftarrow \text{valid}$ ;
34:     end if
35:   end if
36: end for
37: end for
38: Return  $\bar{G}$ 
39:

```

the probability of single tree nodes is the boolean product of probabilities of its parent nodes (indicated by line 8 in Algorithm 1 and shown in (1)).

$$\Pr[\text{Single Tree Node}_i] = \prod \Pr[\text{Parents}(\text{Single Tree Node}_i)] \quad (1)$$

For example, parents of Node 14 are nodes at layer 0 with tree number 2, i.e. Nodes 2, 8, 5, 11. Hence the probability of Node 14 is the product of probabilities of Nodes 2, 8, 5, 11 (shown in (2)).

$$\begin{aligned} \Pr[\text{Node}_{14}] &= \Pr[\text{Node}_2] \times \Pr[\text{Node}_8] \times \Pr[\text{Node}_5] \\ &\quad \times \Pr[\text{Node}_{11}] \\ &= 1 \times 1 \times 0 \times 1 = 0 \end{aligned} \quad (2)$$

Accordingly, if a node belongs to multiple trees (indicated by line 12) (e.g., Node 13), it means that this attack state can exist due to corruption at multiple network locations. Hence the probability of multiple tree nodes (such as Node 13) is the boolean sum of probabilities of parent nodes in different trees (indicated by lines 13–23 in Algorithm 1 and shown in (3)).

Fig. 5 Fragment of marked attack graph for alerting attack

$$\Pr[\text{Multi Tree Node}_i] = \sum_i \text{Parents}_{t_i}[\text{Multi Tree Node}] \quad (3)$$

For example, Node 13, belongs to trees 1, 3, 4, and 7. Hence probability of Node 13, is the sum of probabilities of parent nodes in tree 1, probabilities of parent nodes in tree 3, probabilities of parent nodes in tree 4, and probabilities of parent nodes in tree 7 (shown in (4)).

$$\begin{aligned} \Pr[\text{Node}_{13}] &= \text{Parent}_{t_1} + \text{Parent}_{t_3} + \text{Parent}_{t_4} + \text{Parent}_{t_7} \\ &= (\Pr[\text{Node}_1] \times \Pr[\text{Node}_7] \times \Pr[\text{Node}_3] \times \Pr[\text{Node}_9]) \\ &\quad + (\Pr[\text{Node}_2] \times \Pr[\text{Node}_8] \times \Pr[\text{Node}_3] \times \Pr[\text{Node}_9]) \\ &\quad + (\Pr[\text{Node}_3] \times \Pr[\text{Node}_9] \times \Pr[\text{Node}_4] \times \Pr[\text{Node}_{10}]) \\ &\quad + (\Pr[\text{Node}_3] \times \Pr[\text{Node}_9] \times \Pr[\text{Node}_6] \times \Pr[\text{Node}_{12}]) \\ &= (1 \times 1 \times 1 \times 1) + (1 \times 1 \times 1 \times 1) + (1 \times 1 \times 1 \times 1) \\ &\quad + (1 \times 1 \times 1 \times 1) = 1 \end{aligned} \quad (4)$$

Thus this algorithm computes probabilities of all nodes layer by layer in an iterative manner. With the probabilities computed, the algorithm marks all nodes with 0 probability as invalid and other nodes as valid (indicated by lines 30–34 in Algorithm 1). The algorithm also handles cases where nodes effect two or more trees.

4.2.3 Output

The attack graph shown in Fig. 5 is part of the output obtained from marking the attack graph for the alerting attack. All invalid attack effects (nodes) are shown shaded whereas all valid nodes are shown unshaded. The attack graph shows 7 trees i.e., 7 effects of corruption. By using MAPSec, we can see that 3 of the 7 possible attacks can be eliminated.

4.2.4 Hot-spots

Our algorithm also marks the network *hot-spots* in the attack graph. With respect to the attack graph, we define *hot-spots* as pre-condition nodes i.e. layer 0 nodes, with the highest tree number count. A high tree number count in a pre-condition node indicates an increased attractiveness of the network location to adversaries. This is because by breaking into the network location indicated by the pre-condition node, the adversary has a higher likelihood to succeed, and can cause the highest amount of damage. Our marked attack graphs also show if MAPSec offers any protection to these hotspot nodes.

The number of pre-condition nodes chosen as hot-spots can be set as per user requirements. In our case we set this value to 4. In the case of the alerting attack, the pre-condition nodes indicate the network hot-spots are Nodes 3, and 9. Together these nodes indicate the most attractive physical access, to adversary is at level 3 (from Node 1), i.e. direct access to service nodes by exploitation of data source or service logic vulnerabilities (from Node 3) particularly the MSC service node (from Node 4). By protecting these hot-spot network locations, it is possible to prevent a large number of attacks or have a huge reduction effect on network-wide vulnerabilities. For example, by protecting hot-spots in Fig. 5, attacks presented in trees 1, 3, 4, and 7 can be prevented. By observing the hot-spot pre-condition nodes it is evident that MAPSec does not offer protection to the hot-spot nodes.

Marking of network *hot-spots* is particularly useful in extremely large attack graphs because users are immediately aware of most vulnerable network parts which when protected can prevent a large number of attacks. We now present our coverage measurement formulas which uses the results obtained from the attack graph to quantify the impact of security solutions.

5 Measuring effective coverage of security protocols

The CMF comprises of a set of three formulas to capture the coverage of security protocols. Our triplet formula comprises of (i) *Effective Coverage*, to capture the average effective number of attacks eliminated by the security protocol; (ii) *Deployment coverage*, to capture the coverage of protocol deployments; (iii) *Attack Coverage*, to capture the attack coverage provided by the security protocol. In the following we detail each formula.

1. The *Effective Coverage* is computed as a ratio of number of attacks eliminated by the security protocol to the number of protocol deployments (shown in (5)). A high value of Effective Coverage indicates that the security protocol offers a higher protection for a security deployment.

$$\text{Effective Coverage} = \frac{\# \text{ of Attacks Eliminated}}{\# \text{ of Protocol Deployments}} \quad (5)$$

In the case when MAPSec is applied to the alerting attack (from Fig. 5), the number of attacks prevented is 3 (Trees 2, 5, 6), and the number MAPSec deployment, i.e. MAP messages protected, is 3 (MAP_SRI, MAP_PRN, Page MS). Hence, Effective Coverage when MAPSec is used to protect from alerting attack is 3/3 (shown in (6)).

$$\text{Effective Coverage} = \frac{3}{3} = 1 \quad (6)$$

2. The *Deployment coverage* is computed as a ratio of number of protocol deployments to the number of required deployments (shown in (7)).

$$\begin{aligned} \text{Deployment Coverage} \\ = \frac{\# \text{ of Protocol Deployments}}{\text{Total } \# \text{ of Required Protocol Deployments}} \end{aligned} \quad (7)$$

In the case when MAPSec is applied to the alerting attack (from Fig. 5), the number of protocol deployments i.e. MAP messages protected is 3, and the total number of required deployments i.e. MAP messages requiring protection is 4 (from Table 1, which shows the total number of messages in each MAP service). Hence, Deployment Coverage when MAPSec is used to protect from alerting attack is 3/4 (shown in (8)).

$$\text{Deployment Coverage} = \frac{3}{4} = 0.75 \quad (8)$$

3. The *Attack Coverage* is computed as a ratio of number of attacks eliminated to the total number of attacks in the attack graph (shown in (9)). The higher

Table 1 Total number of message

MAP service name	Total # of MAP messages
Location update service	16
Short message service	14
Handover service	8
Supplementary service	14
Call handling service	4

this value the greater is the security solution's efficacy in eliminating a large number of attacks on the network.

$$\text{Attack Coverage} = \frac{\# \text{ of Attacks Eliminated}}{\text{Total } \# \text{ of Attacks in the Attack Graph}} \quad (9)$$

In the case when MAPSec is applied to the alerting attack (from Fig. 5), the number of attacks prevented is 3, and total number of attacks in the alerting attack scenario is 7. Hence, Attack Coverage when MAPSec is used to protect from alerting attack is 3/7 (shown in (10)).

$$\text{Attack Coverage} = \frac{3}{7} = 0.42 \quad (10)$$

5.1 Interpretation

Typically a good measure of CMF is a low value of deployment coverage, a high value of effective coverage, and a high value of attack coverage. In the case when MAPSec is applied to the alerting attack, a deployment coverage of 75% offers an attack coverage of 42%, and effective coverage of 1. From this we can infer that a single deployment of MAPSec protects against a single attack. Also, a 75% deployment offers an attack coverage of 42% which illustrates the limited effectiveness of MAPSec.

As our formulas consider the effective coverage as a percentage of the network-wide vulnerabilities, they provide a value that reflects the protection coverage of the security protocol on the network-wide vulnerabilities. As our results are generated from attack graphs, which are a product of specifications, we can intuitively argue that our CMF formula is accurate, and gives users an idea of network-wide vulnerabilities.

As MAPSec's performance in the case of the alerting attack did not seem impressive especially with respect to coverage of attacks, hot-spots and their protection, we further evaluate MAPSec with respect to other MAP services before we conclude MAPSec's evaluation.

6 Evaluating MAPSec by marking attack graphs

In this section, we evaluate MAPSec extensively in conjunction with all the major MAP services. We detail our evaluation with respect to the Short Message (SM), Supplementary (SS) Service, and authentication services. In doing so, we detail how eCAT may be applied to real life scenarios to obtain insightful results.

6.1 Short message service

The SM service is used to store and forward short text messages to the subscriber. Subscribers may send short messages (called mobile originated or MO in telecom terminology), or receive short messages (called mobile terminated or MT in telecom terminology). An attack on the SM service may occur as follows. The adversary may corrupt the service center address or 'SC_Address' in the MAP message. As a result the short message may be sent to a rogue service center. Adversaries may benefit by capturing these text messages.

On setting the boolean probability of Node 5 to 0 (because usage of MAPSec can eliminate level 2 access), eCAT produces the resulting marked attack graph shown in Fig. 6. Hot-spots are identified as Nodes 1, 3, and 4. Together these nodes indicate that highest vulnerability of the network exists at level 3 access (from Node 1), i.e. direct access to service nodes by exploitation of data source or service logic vulnerabilities (from Node 3), particularly the MSC service node (from Node 4). By protecting these hot-spot network locations, attacks presented in trees 1, 2, 3, 4, 7, 8, and 9 can be eliminated. Note that MAPSec does not offer protection to these hot-spots.

CMF for SM attacks: From Fig. 6, it can be observed that the number of attacks eliminated is 4 (Trees 5, 6, 10, 11), the number of MAPSec deployments, i.e. MAP messages protected is 3 (MAP_MO_Fwd_SM, MAP_MT_Fwd_SM, MAP_MT_Fwd_SM_ACK), the total number of attacks in the SM attack scenario is 13, and total number of required deployments i.e., MAP messages requiring protection, is 14 (from Table 1). Hence, Effective Coverage when MAPSec is used to protect from SM attack is 1.33, Deployment Coverage is 0.21, and Attack Coverage is 0.31. While MAPSec's coverage is better here than the alerting attack, MAPSec neither has a greater attack coverage nor has a higher impact factor.

6.2 Supplementary service

The Supplementary Service (SS) comprises of additional services that augment the basic service. These include call forwarding, call holding, barring of incoming calls, barring of outgoing calls, call line identification, and so on. The MAP protocol is extensively used by the SS to register, erase, activate, deactivate, and respond to subscriber's queries concerning these services. The SS is denoted by a data item called supplementary service code ('SS_code') which denotes the type of supplementary service being registered, erased, activated, or deactivated. For example, barring of incoming calls is denoted by 'SS_code' of value 351.

By corrupting this 'SS_code', it is possible to register a subscriber for a different service i.e., if a subscriber wants to register for call forwarding, they could be registered for barring incoming calls. Using eCAT to detect all possible effects of 'SS_code' corruption at various network locations, we have constructed an attack graph (a fragment of

Fig. 6 Fragment of marked attack graph for SM attacks

Fig. 7 Fragment of marked attack graph for SS attacks

which is shown in Fig. 7). This attack graph shows effects of corruption of 'SS_code' at possible locations.

On setting the boolean probability of Node 14 to 0 (because usage of MAPSec can eliminate level 2 access), eCAT produces the marked attack graph shown in Fig. 7. Hot-spots are identified as Nodes 7, 8, and 9. Together these nodes indicate that highest vulnerability of the network exists at level 3 access (from Node 9), i.e. direct access to service nodes by exploitation of data source or service logic vulnerabilities (from Node 7), particularly the MSC service node (from Node 8). By protecting these hot-spot network locations, attacks presented in trees 1, 2, 3, 5, 6, 7, 8, 9, and 10 can be prevented. Once again, note that MAPSec does not offer protection to these hot-spots.

CMF for SS attacks: From Fig. 7, it can be observed that MAPSec protects 3 MAP messages 'MAP_deactivate_SS' arriving at VLR, 'MAP_deactivate_SS' arriving at HLR, and 'MAP_Register_SS'. Also, a typical supplementary service comprises of 14 messages (obtained from Table 1). Hence, *Deployment Coverage* when MAPSec is used to protect supplementary service is 0.21. We can also observe from Fig. 8, that MAPSec prevents 3 out of 14 attacks. Therefore, *Attack Coverage* is 0.21, and *Impact factor* is 1. Hence for a deployment coverage of 21%, MAPSec has an attack reduction of 21% and an impact factor of 1.

6.3 Authentication service

The MAP protocol carries authentication material as clear text in its messages. This material is used to authenticate subscribers and allows them to access location update, handover, short message services, and so on. The authentication material is stored by the HLR in the form of a triplet array of random values (RAND), signature of random value (SRES), and ciphering keys (K_c). When a subscriber registers or moves to a new location, the HLR provides the corresponding VLR with this triplet array

using MAP messages. The VLR picks a random index for RAND and passes this RAND value to the subscriber's mobile device. The mobile device then uses the RAND value to compute the SRES and K_c . The index value of the cipher key K_c is called Ciphering Key Sequence Number or 'CKSN'.

If the RAND value is corrupt, the subscriber's mobile will compute an incorrect SRES which it uses to authenticate itself to the VLR. The VLR finds the SRES to be incorrect and refuses service to the subscriber. The subscriber and the network communicate over the air interface by encrypting their communications using one of the cipher keys K_c . Also, during lengthy network connections the network may re-negotiate the keys. To ensure that the subscriber and the network are using the same cipher keys K_c , the CKSN is sent using MAP messages. If the CKSN is incorrect, the subscriber cannot be understood and hence the communication is ineffective. Thus, attacks on authentication material can effect all the services.

We use eCAT to detect possible attacks due to corruption of 'RAND' and 'CKSN' corruption at various network locations. A fragment of the attack graph shown in Fig. 8. On setting the boolean probability of Node 7 to 0 (because usage of MAPSec can eliminate level 2 access), eCAT produces the resulting marked attack graph shown in Fig. 8. Hot-spots are identified as Nodes 4, 5, and 6. Together these nodes indicate that highest vulnerability of the network exists at level 3 access (from Node 4), i.e. direct access to service nodes by exploitation of data source or service logic vulnerabilities (from Node 6), particularly the VLR service node (from Node 5). By protecting these hot-spot network locations, attacks presented in trees 1, 2, 3, 5, 6 and 7 can be prevented. Note, once again that MAPSec does not offer protection to these hot-spots.

CMF for authentication material attacks: From Fig. 8, we observe that MAPSec protects MAP messages 'MAP_Send_Id_Response', 'MAP_Authenticate', and 'MAP_Prepare_HO'. Also, the total of all MAP messages

Fig. 8 Fragment of marked attack graph for authentication material attacks

considered in this scenario is 38 (Table 1), including the location update, handover and short message services. Hence, *Deployment Coverage* when MAPSec is used to protect authentication attacks is 0.07. As MAPSec prevents 3 out of 10 attacks, therefore, *Attack Coverage* is 0.3, and *Effective coverage* is 1. Hence, although the impact factor seems to be low, a low deployment coverage of MAPSec can reduce attacks by 30%. Hence deploying MAPSec in this case may be adequate if not extremely effective.

6.4 Experiments and analysis

To further confirm the coverage provided by MAPSec, we have conducted extensive experiments and analysis. We evaluated MAPSec's performance in all the major services that could use MAPSec including—*location update, short message, handover, supplementary and call handling services*.

In our testing, we considered each individual MAP message in each service and found the number of attacks that may be prevented by protecting that individual MAP message using MAPSec. Our extensive test results are documented in Table 2A–E.

In each table, the first column is the name of the MAP message protected by MAPSec. The second column is the number of attacks that may be eliminated by protecting the corresponding MAP message with MAPSec. The third column is the total number of attacks that can occur without MAPSec's protection; this typically corresponds to all the attack trees in the attack graph. These include attacks due to data source and service logic corruption, as well as attacks due to corruption of data items in MAP messages. This column gives us the network-wide attacks. The final three columns correspond to values of effective coverage, deployment coverage, and attack coverage for protecting the corresponding MAP message using MAPSec. The final row corresponds to the values for protecting all the MAP messages in the service. The following are the observations from our tables.

- A major observation from our tables is that protecting a single MAP message or a selected set of MAP messages is better than protecting all the MAP messages in a service. For example, consider the supplementary service. Protecting a single message 'MAP_DEACTIVATE_SS_ACK' gives better values of effective coverage, deployment coverage, and attack coverage (3, 0.07, 0.375) over protecting all the MAP messages (2.85, 1, 0.29) in the supplementary service.
- Other observations include: (i) MAPSec has an average network-wide attack coverage of 33%, with a maximum attack coverage of 65%, and a minimum attack coverage of 0%; and (ii) the effective coverage from deploying MAPSec can be a high of 26 to a low of 0.
- Most vulnerable network areas, i.e. hot-spots corresponds to data sources and service logic, are not protected by MAPSec. Hence the other security protocols required are for protecting data sources and service logic.

In the next section, we detail further observations from our tables.

7 MAPSec utility findings and discussions

In this section, we qualitatively discuss our findings from evaluating MAPSec. Our findings may be categorized as with respect to the network in general and with respect to MAPSec.

7.1 Data sources and service logic

Data Sources and Service Logic are the most vulnerable network areas. This is because a corrupt data source or service logic may be used by many different services and hence, cause many varied cascading effects spawning a large number of attacks (indicated by multiple trees in attack graphs). Thus attacks that occur due to exploiting data source vulnerabilities constitute a major portion of the

Table 2 Experimental results

A. LOCATION UPDATE SERVICE

MAP_Message_Name	# Attacks prevented	Total # of possible attacks	Effective coverage	Deployment coverage	Attack coverage
MAP_UPDATE_LOCATION_AREA	2	8	2	0.06	0.25
MAP_UPDATE_LOCATION_AREA-ACK	1	4	1	0.06	0.25
MAP_SEND_IDENTIFICATION	3	12	3	0.06	0.25
MAP_SEND_IDENTIFICATION-ACK	3	13	3	0.06	0.23
MAP_SET_CIPHERING_MODE	2	8	2	0.06	0.25
MAP_SET_CIPHERING_MODE-ACK	0	0	0	0.06	0
MAP_FORWARD_NEW_TMSI	1	4	1	0.06	0.25
MAP_FORWARD_NEW_TMSI -ACK	0	0	0	0.06	0
MAP_UPDATE_LOCATION	8	30	8	0.06	0.27
MAP_UPDATE_LOCATION-ACK	2	8	2	0.06	0.25
MAP_CANCEL_LOCATION	3	9	3	0.06	0.33
MAP_CANCEL_LOCATION-ACK	0	0	0	0.06	0
MAP_INSERT_SUBSCRIBER_DATA	26	40	26	0.06	0.65
MAP_INSERT_SUBSCRIBER_DATA-ACK	6	18	6	0.06	0.33
MAP_AUTHENTICATE	2	6	2	0.06	0.33
MAP_AUTHENTICATE-ACK	1	3	1	0.06	0.33
All Messages	60	163	3.75	1	0.37

B. SHORT MESSAGE SERVICE

MAP_Message_Name	# Attacks prevented	Total # of possible attacks	Effective coverage	Deployment coverage	Attack coverage
MAP-SEND-ROUTING-INFO-FOR-SM	6	20	6	0.07	0.3
MAP-SEND-ROUTING-INFO-FOR-SM-ACK	5	14	5	0.07	0.36
MAP-MT-FORWARD-SHORT-MESSAGE	4	12	4	0.07	0.33
MAP-MT-FORWARD-SHORT-MESSAGE-ACK	1	4	1	0.07	0.25
MAP-SEND-INFO-FOR-MT-SMS	1	4	1	0.07	0.25
MAP-SEND-INFO-FOR-MT-SMS-ACK	1	4	1	0.07	0.25
MAP_PROCESS_ACCESS_REQUEST	8	25	8	0.07	0.32
MAP_SEARCH_FOR_MS	1	2	1	0.07	0.5
MAP_AUTHENTICATE	2	6	2	0.07	0.33
MAP_AUTHENTICATE ACK	1	3	1	0.07	0.33
MAP-SEND-INFO-FOR-MO-SMS	1	2	1	0.07	0.5
MAP-SEND-INFO-FOR-MO-SMS-ACK	1	2	1	0.07	0.5
MAP-MO-FORWARD-SHORT-MESSAGE	4	11	4	0.07	0.36
MAP-MO-FORWARD-SHORT-MESSAGE-ACK	1	2	1	0.07	0.5
All Messages	37	111	2.64	1	0.33

C. SUPPLEMENTARY SERVICE

MAP_Message_Name	# Attacks prevented	Total # of possible attacks	Effective coverage	Deployment coverage	Attack coverage
MAP_REGISTER_SS	8	25	8	0.07	0.32
MAP_REGISTER_SS_ACK	3	11	3	0.07	0.27
MAP_ERASE_SS	2	6	2	0.07	0.33
MAP_ERASE_SS_ACK	2	7	2	0.07	0.285

Table 2 continued

C. SUPPLEMENTARY SERVICE

MAP_Message_Name	# Attacks prevented	Total # of possible attacks	Effective coverage	Deployment coverage	Attack coverage
MAP_ACTIVATE_SS	3	10	3	0.07	0.3
MAP_ACTIVATE_SS_ACK	4	14	4	0.07	0.285
MAP_DEACTIVATE_SS	2	6	2	0.07	0.33
MAP_DEACTIVATE_SS_ACK	3	8	3	0.07	0.375
MAP_INTERROGATE_SS	3	9	3	0.07	0.33
MAP_INTERROGATE_SS_ACK	7	20	7	0.07	0.35
MAP_INVOKE_SS	2	6	2	0.07	0.33
MAP_INVOKE_SS_SS_ACK	0	0	0	0.07	0
MAP_GET_PASSWORD_SS	0	0	0	0.07	0
MAP_GET_PASSWORD_SS_ACK	1	3	1	0.07	0.33
All Messages	40	135	2.85	1	0.29

D. CALL HANDLING SERVICE

MAP_Message_Name	# Attacks prevented	Total # of possible attacks	Effective coverage	Deployment coverage	Attack coverage
MAP_SEND_ROUTING_INFORMATION	23	67	23	0.25	0.34
MAP_SEND_ROUTING_INFORMATION_ACK	16	49	16	0.25	0.326
MAP_PROVIDE_ROAMING_NUMBER	18	50	18	0.25	0.36
MAP_PROVIDE_ROAMING_NUMBER_ACK	2	8	2	0.25	0.25
All Messages	59	174	14.75	1	0.34

E. HANDOVER SERVICE

MAP_Message_Name	# Attacks prevented	Total # of possible attacks	Effective coverage	Deployment coverage	Attack coverage
MAP_PREPARE_HANDOVER-REQ	9	36	9	0.125	0.25
MAP_PREPARE_HANDOVER-RES	6	22	6	0.125	0.27
MAP_PROCESS_ACCESS-SIGNALLING	5	20	5	0.125	0.25
MAP_SEND_END_SIGNAL-REQ	4	16	4	0.125	0.25
MAP_SEND_END_SIGNAL-RES	3	9	3	0.125	0.33
MAP_SEND_HANDOVER-REPORT-REQ	2	8	2	0.125	0.35
MAP_SEND_HANDOVER-REPORT-RES	0	0	0	0.125	0
MAP_ALLOCATE_HANDOVER-NUMBER	0	0	0	0.125	0
All Messages	29	111	3.62	1	0.26

network-wide vulnerabilities and hence a major problem. In other words by exploiting service logic and data sources the likelihood of attack success is very high. Hence data source and service logic protection mechanisms must be deployed.

It must be noted that MAPSec protects neither service logic nor data sources, rather it protects MAP messages (level 2). Message corruption has a low spawning effect. Typically a single message corruption causes a single attack, as messages are typically used by a single service.

As a result, MAPSec is a solution to a minor portion of the security problem, and only protects up to 33% (from our experiments) of attacks.

7.2 Lack of protection from cascading effects

Another observation is that, although MAPSec can prevent MAP message attacks from occurring, MAPSec cannot prevent a successfully launched attack from cascading.

When an incoming MAP message (unprotected by MAPSec) is corrupt, the service node assumes it is correct, and hence securely transports it, thereby aiding cascading effects. For MAPSec to be truly successful every leg of the MAP message transport must be secured using MAPSec.

7.3 Risk level of MAP message attacks

Finally, the risk level of attacks without MAPSec is very high. MAP messages can be very easily corrupted when they traverse third party networks enroute to their destinations. Rogue third parties can very easily target messages from a certain source, or to certain destinations. While the risk level of service logic or data source corruption may be low because service providers typically house service nodes in central offices where they have some measure of control over direct access to physical equipment, this may not be possible when messages traverse through third party networks.

In conclusion, MAPSec is 100% true to its threat model and can prevent the occurrence of all MAP message related attacks which are extremely high risk. As MAPSec's deployment is optional, if any service provider chooses to omit its deployment, the efforts of all other providers is wasted. Hence to completely protect MAP messages, MAPSec's deployments must be used by every service provider. However the overhead for deploying MAPSec can be high, both in terms of processing load and monetary investment.

Finally, as MAPSec can only protect 33% of attacks, it alone is insufficient to protect the network. A complete protection scheme for the network must comprise of data source and service logic protection in addition to MAPSec.

8 Related work

In this section, we review the literature on various attack graph technologies and illustrate that our method is the first to conduct *security protocol coverage analysis for mobile telecommunication networks*. Attack graph technologies have been extensively studied by [12–14]. Each of these techniques show how attacks can occur, and progress by exploiting vulnerabilities. While a lot of research has been conducted in uncovering attacks using graphs, few have used graphs for practice oriented analysis. In the following, we summarize major attack graph technologies that have used graphs for further analysis.

Amman et al. [15, 17], and Swiler and Phillips [12–14] have developed schemes to search their attack graphs to find the shortest and the most likely to succeed attack paths. Our scheme is similar to theirs, as we have also developed a scheme to find attacks, from our attack graph, that can be prevented using MAPSec. However, the focus of our work is entirely different, in the sense that we want

to evaluate a security scheme by quantifying its coverage, and its effectiveness.

Jha et al. [16, 19, 20] used model checking to generate attack graphs. They analyzed their graphs for survivability, reliability, and minimization. In each case they attempt to solve the 'minimum cost and maximum benefit' problem, for example, finding the minimum critical set of atomic attacks that must be prevented to guarantee that the adversary cannot reach the goal. They approach the problem by considering attack likelihoods to be *markov decision processes* (MDP) and used the value iteration algorithm to find the optimal selection policy. Although our work is similar in aim, their solution cannot be applied to our case, as the markov process assumption is inapplicable to mobile telecommunication networks. In detail, markov processes tend to approximate the network to a single state, but the mobile telecommunication network is a concurrent distributed system, whose current state is the collective state of all its service nodes and messages.

Noel et al. [30] use dependency graphs to compute a set of minimum cost network hardening measures to guarantee the safety of the system. The solution is a set of initial conditions that can be independently disabled at minimum cost. Although the focus of our work is similar i.e. get maximum protection, nevertheless our work is more practice oriented in the sense that: (1) Our approach is focused on measuring the effectiveness of security protocols in terms of the attacks it can protect; (2) We have provided a formula to quantify the same; and (3) We evaluate the performance of a MAPSec against real network.

To the best of our knowledge our work is the first of its kind to analyze deployment impact of security solutions on mobile telecommunication networks, provide a CMF formulas, and most importantly evaluate MAPSec and provide insights into its utility.

9 Conclusion

We have evaluated MAPSec using a highly effective graph based scheme—eCAT. Results from the evaluation of MAPSec have shown that while MAPSec provides protection from a narrow set of attacks, it does not effectively mitigate attacks with the largest potential impact, i.e. attacks caused by corrupt data sources, and service logic. In fact, we have shown that attacks that have previously occurred would not be prevented by the current versions of MAPSec. The technical impact of this research is that, it is now possible to uncover vulnerabilities in mobile telecommunication networks, determine their potential impact, and evaluate solutions to fix them.

Acknowledgements Peng Liu was supported in part by NSF ANI-0335241.

References

1. Kotapati, K., Liu, P., Sun, Y., & La Porta, T. F. (2005). A taxonomy of cyber attacks on 3G networks. In *Proceedings IEEE International Conference on Intelligence and Security Informatics, ISI*, Lecture Notes in Computer Science (pp. 631–633). Springer-Verlag, May 2005.
2. Kotapati, K., Liu, P., & La Porta, T. F. (2007). Dependency relation based vulnerability analysis of 3G networks: Can it identify unforeseen cascading attacks? *Special Issue of Springer Telecommunications Systems on Security, Privacy and Trust for Beyond 3G Networks*, Mar 2007.
3. Kotapati, K., Liu, P., & La Porta, T. F. (2006). CAT—a practical graph & SDL based toolkit for vulnerability assessment of 3G networks. In *Proceedings of the 21st IFIP TC-11 International Information Security Conference, "Security and Privacy in Dynamic Environments"*, SEC 2006, May 2006.
4. Chatras, B., & Vernhes, C. (1990). Mobile application part design principles. In *Proceedings of XIII International Switching Symposium* (Vol. 1, pp. 35–42), June 1990.
5. Audestad, J. A. (2004). The mobile application part (map) of gsm. Technical Report Telektronikk 3.2004, Telektronikk, Mar 2004.
6. 3GPP. Mobile application part (map) specification. Technical Standard 3GPP TS 29.002 V3.4.0, 3G Partnership Project, Apr 1999.
7. Audestad, J. A. (1988). Network aspects of the gsm system. In *Proceedings of EUROCON 88*, June 1988.
8. Telcoman. CENTRAL OFFICES. In <http://www.thecentraloffice.com/>.
9. Switch. SESS Switch. In <http://www.alleged.com/telephone/SESS/>.
10. Prevelakis, V., & Spinellis, D. (2007). The Athens affair. *IEEE Spectrum*, July 2007.
11. 3GPP2 3GPP. Third Generation Partnership Project. In <http://www.3gpp.org/>.
12. Swiler, L. P., Phillips, C., & Gaylor, T. (1998). A graph-based network vulnerability analysis system. SandiaReport SAND97-3010/1, Sandia National Laboratories, Jan 1998.
13. Phillips, C., & Swiler, L. P. (1998). A graph-based system for network-vulnerability analysis. In *Proceedings of the 1998 Workshop on New Security Paradigms (NSPW '98)*.
14. Swiler, L., Phillips, C., Ellis, D., & Chakerian, S. (2001). Computer-attack graph generation tool. In *Proceedings of the DARPA Information Survivability Conference and Exposition II*, June 2001.
15. Ritchey, R. W., & Ammann, P. (2000). Using model checking to analyze network vulnerabilities. In *Proceedings 2000 IEEE Computer Society Symposium on Security and Privacy* (Vol. 00, pp. 156–165), May 2000. Los Alamitos: IEEE Computer Society.
16. Sheyner, O., Haines, J., Jha, S., Lippmann, R., & Wing, J. M. (2002). Automated generation and analysis of attack graphs. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, May 2002.
17. Ammann, P., Wijesekera, D., & Kaushik, S. (2002). Scalable, graph-based network vulnerability analysis. In *Proceedings of the 9th ACM conference on Computer and communications security (CCS '02)* (pp. 217–224), Nov 2002.
18. Ou, X., Govindavajhala, S., & Appel, A. W. (2005). MulVAL: A logic-based network security analyzer. In *Proceedings of the 14th Usenix Security Symposium* (pp. 113–128), Aug 2005.
19. Jha, S., Sheyner, O., & Wing, J. (2002). Two formal analyses of attack graphs. In *Proceedings of the 15th IEEE Computer Security Foundations Workshop (CSFW'02)* (p. 49), June 2002. Washington: IEEE Computer Society.
20. Jha, S., Sheyner, O., & Wing, J. M. (2002). Minimization and reliability analyses of attack graphs. Technical Report CMU-CS-02-109, Feb 2002.
21. Sheyner, O., & Wing, J. (2005). Tools for generating and analyzing attack graphs. In *Proceedings of Formal Methods for Components and Objects*, Lecture Notes in Computer Science (pp. 344–371).
22. Templeton, S. J., & Levitt, K. (2000). A requires/provides model for computer attacks. In *Proceedings of the 2000 workshop on New security paradigms (NSPW '00)* (pp. 31–38). New York: ACM Press.
23. Cuppens, F., & Miège, A. (2002). Alert correlation in a cooperative intrusion detection framework. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy* (pp. 202–215), May 2002.
24. Zerkle, D., & Levitt, K. (1996). NetKuang—a multi-host configuration vulnerability checker. In *Proceedings of the Sixth USENIX Security Symposium* (pp. 195–201).
25. Cuppens, F., & Ortalo, R. (2000). Lambda: a language to model a database for detection of attacks. In *Recent Advances in Intrusion Detection* (pp. 197–216).
26. Ramakrishnan, C. R., & Sekar, R. C. (2002). Model-based analysis of configuration vulnerabilities. *Journal of Computer Security*.
27. Ning, P., Cui, Y., & Reeves, D. S. (2002). Constructing attack scenarios through correlation of intrusion alerts. In *Proceedings of the 9th ACM Conference on Computer & Communications Security (CCS '02)* (pp. 245–254), Nov 2002.
28. Ning, P., & Xu, D. (2003). Learning attack strategies from intrusion alerts. In *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS '03)* (pp. 200–209), Oct 2003.
29. Ning, P., Xu, D., Healey, C. G., & St. Amant, R. A. (2004). Building attack scenarios through integration of complementary alert correlation methods. In *Proceedings of the 11th Annual Network and Distributed System Security Symposium (NDSS '04)* (pp. 97–111), Feb 2004.
30. Noel, S., Jajodia, S., OBerry, B., & Jacobs, M. (2003). Efficient minimum-cost network hardening via exploit dependency graphs. In *Proceedings of the 19th Annual Computer Security Applications Conference*, Dec 2003.

Author Biographies

Kameswari Kotapati received her Bachelor of Engineering degree from the University of Madras at Chennai, India, and her M. S. degree from University of Massachusetts, Dartmouth. She is a Ph.D. candidate in the Department of Computer Science and Engineering at the Pennsylvania State University. Her research interests include signaling protocols and security for mobile telecommunication networks, wireless networks,

and location service security. Her research is in the area of security in emerging wireless telecommunication networks. She is one of the first to perform extensive studies in this area. Her dissertation titled Assessing Security of Mobile Telecommunication Networks has been used as a major input by the Vulnerabilities Threat Modeling Working Group of the Next Generation Networks Task Force of the Presidents National Security Telecommunications Advisory Panel.

Peng Liu received his B.S. and M.S. degrees from the University of Science and Technology of China, and his Ph.D. degree from George Mason University in 1999. Dr. Liu is an associate professor of Information Sciences and Technology and director of the Cyber Security Lab at Penn State. His research interests are in all areas of computer and network security. He is the founding program co-chair of the ACM Workshop on Survivable and Self-Regenerative Systems. He was the proceed-

ings chair of ACM Conference on Computer and Communications Security (CCS) for 2004 and 2003. He is a program committee member of over thirty-five international conferences and workshops, including ACM Conference on Computer and Communications Security (CCS), INFOCOM, European Symposium on Research in Computer Security (ESORICS), and the World Wide Web Conference (WWW). He is a referee for over twenty journals, including the ACM Transactions on Information and Systems Security and the Journal of Computer Security. He is on the editorial board of Elsevier Computer Standards & Interfaces Journal. Dr. Liu has published a book and about 80 refereed technical papers. His research has been sponsored by DARPA, NSF, DOE, DHS, AFRL, NSA, CISCO, HP, Japan JSPS, and Penn State. Dr. Liu is a recipient of the DOE Early CAREER PI Award. More information about Peng Liu can be found at <http://www.ist.psu.edu/s2>.

Thomas F. La Porta received his B.S.E.E. and M.S.E.E. degrees from The Cooper Union, New York, NY, and his Ph.D. degree in Electrical Engineering from Columbia University, New York, NY. He is a Distinguished Professor in the Computer Science and Engineering Department at Penn State. He is the director of the Networking Research Center at Penn State. Prior to joining Penn State, Dr. La Porta was with Bell Laboratories

since 1986. He was the director of the Mobile Networking Research Department in Bell Laboratories, Lucent Technologies where he led various projects in wireless and mobile networking. He is an IEEE Fellow, Bell Labs Fellow, received the Bell Labs Distinguished Technical Staff Award in 1996, and an Eta Kappa Nu Outstanding Young Electrical Engineer Award in 1996. His research interests include mobility management, signaling and control for wireless networks, mobile data systems, and protocol design. Dr. La Porta was the founding Editor-in-Chief of the IEEE Transactions on Mobile Computing, and an associate editor for the ACM/Kluwer Journal of Mobile Networking and Applications as well as the KICS Journal of Communications and Networks. He served as editor-in-chief of IEEE Personal Communications Magazine (now called IEEE Wireless Communications) for 3 years and is currently a senior advisor. He has published over 50 technical papers and holds 28 patents. He was an adjunct faculty member at Columbia University for 7 years where he taught courses on mobile networking and protocol design.