Appendix A. User Data's Attributes, Spatiotemporal Characteristics, and Spatiotemporal mapping Characteristics

Based on communication processes and traffic provision mechanisms in existing mobile communication networks (e.g., GSM, CDMA, WCDMA, and the like), we excavate different use data' inherent attributes (indicated by A) and diversified functions (indicated by F) in various application scenarios and in different network entities. Here, we treat application scenarios as time domains and network entities as space domains. We also analyze the mapping relationships among different user data items in different time-space points.

User data's inherent attributes include user data's definition, structure, source, and to name just a few. User data's functions, according to a wide range of mobile application scenarios, may be identity authorization functions, routing and addressing functions, billing functions, calling line identification functions, locating functions, and so on. We define three mapping relationships, namely, forward mapping, reverse mapping, and bidirectional mapping. Given two data items A and B, A is forward mapping to B if B is capable of being retrieved according to A; A is reverse mapping to B if A is capable of being accessed according to B; A is bidirectional mapping to B if A and B can be retrieved according to each other.

Then, we analyze user data's roles (denoted by R) in different time-space points. We model user data's roles using dominant (denoted by D), auxiliary (denoted by A), and optional (denoted by O) in specific time-space domain. Different user data play distinct roles in a same time-space point; the same user data may play different roles when time-space point is changing. For example, in a scenario where user A calls user B, to obtain B's traffic routing, B's MSISDN number is needed by A's serving MSC/VLR to find B's HLR. In this case, B's MSISDN number is playing a dominant role. B's IMSI and location identification, performing the billing functions, are playing auxiliary roles. The existing mobile communication mechanisms ensure that no user data is allowed to be changed if the user data are in dominant roles.

With a thorough analysis of use data's inherent attributes and their various functions, we can determine user data's roles and their mapping relationships in the time–space domains. We call user data's function and role spatiotemporal-characteristic and call user data's mapping-relationship spatiotemporal-mapping-characteristic. Figure A-1 depicts multiple user data's attributes, spatiotemporal characteristics and spatiotemporal mapping characteristics.

In Figure A-1, d_x(x=a, *b*, *c*…) infers different data item. We observe from Figure A-1 that the spatiotemporal relationship sequences can be obtained from user data's spatiotemporal characteristics and mapping characteristic. It is noteworthy that the spatiotemporal relationship sequences intuitively indicate user data's attributes, functions, roles and mutual mapping relation in corresponding application scenarios and network entities.

The spatiotemporal relationship sequences offer a general guideline for us to address the challenge issue of whether or not user data can be dynamically manipulated by the DVM mechanism in specific time- space points.

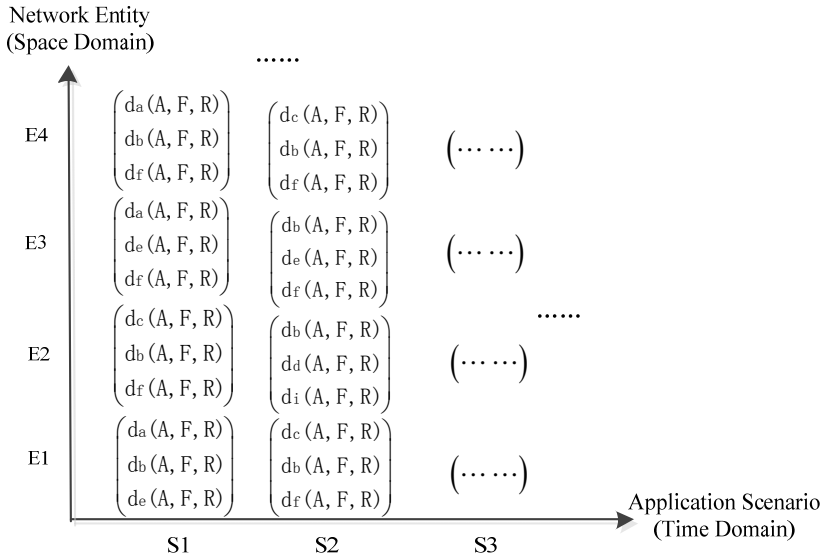


Figure A-1 User data's attributes, spatiotemporal characteristics and spatiotemporal mapping characteristics

Figure A-2 demonstrates the spatiotemporal relationship sequences marked with user data's source attribute (see items labeled in the brackets). Figure A-2 shows that there are six potential sources of the user data: (1) terminal equipment (i.e., T), (2) HLRs (i.e., H), (3) the caller's serving MSC/VLR (i.e., V1), and (4) the callee's serving MSC/VLR (i.e., V2), (5) the local

database of current network entity (i.e., O), and (6) assigned temporarily by the current network entity(i.e., S). The caller refers to the user who initiates a call or some other services; the callee refers to the user who establishes a communication link with the caller.

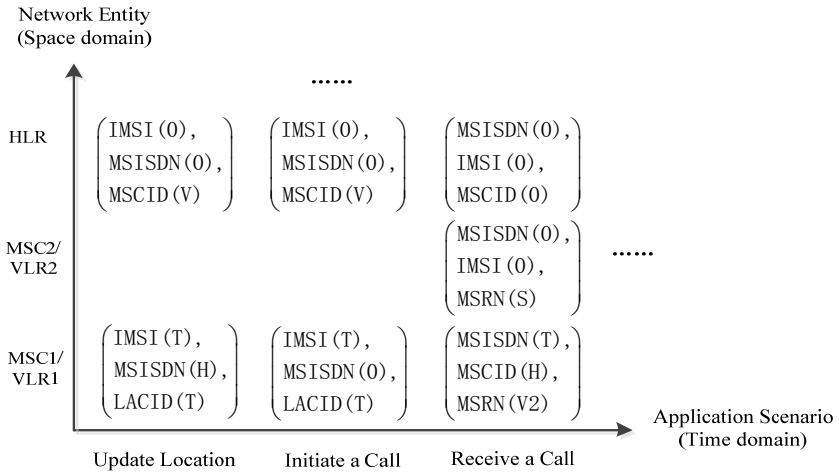


Figure A-2 User data spatiotemporal relationship sequences.

Now we use the user-receiving-a-call scenario (see Figure A-2) as an example to illustrate the time and conditions under which our DVM mechanism is allowed to dynamically manipulate the user data.

In the user-receiving-a-call scenario illustrated in Figure A-2, the callee's MSISDN number is forward mapping to its location identification (MSCID) in MSC1/VLR1, where MSISDN number is the number dialed by the caller, and MSISDN number's source is the caller's terminal device (i.e., labeled by local creation), MSCID is the callee's location identification, and MSCID's source in this scenario is the callee's HLR (labeled by remote creation relative to MSC1/VLR1). Note that when the callee is moving, location identification MSCID is updating correspondingly (labeled by dynamic changeability).

MSISDN number, as a dialed number, with the function of addressing the callee's HLR, is in a dominant role, which does not satisfy the condition of being dynamically manipulated. MSCID, with the billing function, is in a non-dominant role. Otherwise, MSCID satisfies the condition of remote creation and dynamic changeability. Hence, location identification MSCID is capable of being dynamically manipulated in in MSC1/VLR1 and in user-receiving-a-call scenario.

With the dynamic manipulation of MSCID in place, a virtual mapping is established between the callee's MSISDN number and its MSCID in the caller's serving MSC/VLR (i.e., MSC1/VLR1). Such a virtual mapping successfully hides the callee's physical location.

By the same token, we can infer that in other application scenarios, user's MSISDN number can also be dynamically manipulated in a way to hide user locations, routes, IMSI, and other security sensitive information.

We are still facing the challenges of (1) how to implement user data dynamic manipulation, and (2) how to ensure normal communication after user data are dynamically manipulated. To address these two challenges, we have developed a DVM principle prototype based on existing mobile communication networks. In the DVM principle prototype, we verify the dynamic manipulation to location identification (i.e., MSCID) and routing identification (i.e., MSRN), as well as MSISDN number separately in user-receiving-a-call scenario and in update-location scenario. To ensure normal communication, two function modules named by DVM-HLR-partner and DVM-location-agency are deployed. The DVM-HLR-partner keeps all user data's real mapping-relationship list; the DVM-location-agency assists mobile networks in performing normal call-connection. We design secure internal interface for the two function modules. Due to space limitation, we do not present the implementation details.

Appendix B, Security efficiency evaluation

In this section, we try to give some guidance suggestions about how to achieve better security efficiency by adopting the DVM mechanism.

We define the following four parameters coupled with the assumptions made in this study. These parameters allow us to provide a thorough analysis on the DVM's security properties.

(1) Necessary resource acquisition time interval (T_r): This interval T_r represents the average time cost by attackers to ac-

quire a group of necessary resources, or attackers' average time spent prior to reaching the next state from one state of attack chains. We assume that the average time cost to acquire different group of necessary resources is the same.

(2) User data dynamic-manipulation time interval (T_d): This time interval T_d denotes a window within which user data are dynamically manipulated. It is assumed that the time interval keeps fixed once being set by systems; the current dynamic-manipulation process is independent with the previous processes.

(3) dynamic-manipulation occurrence probability (p_{ai}): p_{ai} represents a probability that user data included in R_i are dynamically manipulated during a dynamic-manipulation period. Manipulating user data or not is determined by application scenarios as well as security policies. p_{ai}=0 indicates that the user data included in R_i will not be manipulated during the current dynamic-manipulation period.

(4) p'_i (i=1，2，…，n): p'_i is the probability that attackers have successfully transferred from state S_{i-1} to state S_i on the condition of the DVM mechanism being adopted.

In what follows, we analyze the attack-success probability when the DVM mechanism is adopted.

Under the condition that attackers have acquired necessary resource R_1 and have transferred from initial state S_0 to state S_1, R_1 will not be valid if all or part of user data included in R_1 are manipulated during the time period of T_r. Then, attackers have to transfer back to state S_0 to reacquire R_1. Similarly, under the condition of attackers having acquired necessary resource R_i（$2 \le i \le n$） and having transferred from state S_{i-1} to S_i , R_i will become invalid if some user data included in R_i are manipulated during the time period of T_r. Invalid R_i forces attackers to transfer from state S_i back to state S_{i-1} to reacquire R_i.

If user data included in more than one group of necessary resources have been manipulated concurrently, attackers have to transfer the state to an earlier state in which the necessary resources can be reacquired. In doing so, we guarantee that necessary resources have to be acquired in a serial way. For example, if user data included in R_1 and R_2 are manipulated at the same time, then attackers have to transfer to state S_0.

Figure B-1 depicts the attack state transfer diagram of the DVM mechanism. P_{ij} in Figure B-1 denotes the probability that the state is transferred from S_i to S_j , $i>j$, i=1,2,… ,n, j=0,1,2,…,n-1.

According to the attack processes as well as the attack state transfer diagram, we conclude that attackers' attack state at time instance t only depends on the state at time t-1; the state at time t is independent of any state prior to time t-1. Thus, we model the attack chain as a Markov chain.

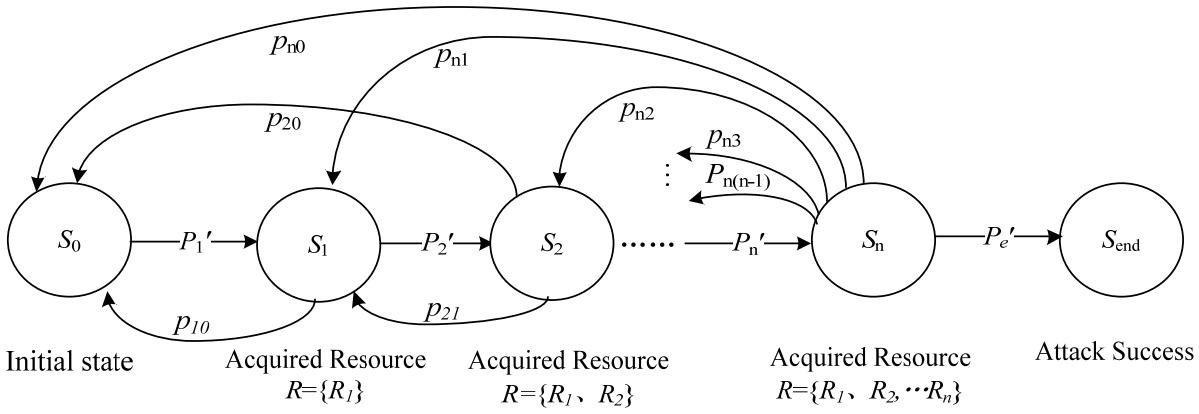


Figure B-1 Attack state transfer diagram of the DVM mechanism.

Let p_{si} （i=1,2,...,n）and p_{se} respectively represent the steady-state probability of attackers' attack state being in S_i and S_{end}. Then, attackers' attack-success probability (i.e., p'_{succ}) is expressed as formula (2).

$$p'_{succ} = p_{se} = p_{sn} * p'_e \qquad (2)$$

In what follows, we analyze the expressions of p_{ij}, p'_i as well as p'_e, which are the state transfer probabilities of the Markov Chain plotted in Figure B-1.

p_{i0} indicates the state transfer probability that user data included in R_1 are manipulated when attackers stay in attack state S_i(i=1,2,...,n). Recall that p_{a1} is the probability that user data included in R_1 are manipulated in the time interval T_d; hence, the probability of user data included in R_1 not being manipulated in the time interval T_d is 1-p_{a1}. There is T_r / T_d times dynamic-manipulation happening during the attack time interval T_r. Therefore, we derive the probability of user data included in R_1 not being manipulated in period of T_r from p_{a1} and T_r / T_d as $(1 - p_{a1})^{Tr/Td}$. Hence, the probability of user data included in

R_1 being manipulated during the attack time interval T_r is $1-\left(1-p_{a1}\right)^{Tr/Td}$. Thus, we obtain p_{i0} as (3).

$$p_{i0} = 1-(1-p_{a1})^{\frac{Tr}{Td}} \tag{3}$$

p_{i1} indicates the state transfer probability that user data included in R_1 are not manipulated when user data included in R_2 are manipulated under the condition of attackers staying in attack state S_i（i=2,...,n）. That is, p_{i1} is expressed as formula (4).

$$p_{i1} = \left(1-P_{a1}\right)^{\frac{Tr}{Td}} * \left(1-\left(1-p_{a2}\right)^{\frac{Tr}{Td}}\right) \tag{4}$$

By the same token, we derive probability p_{ij}（j<i, i=3,…,n; j=2,3,…,n-1,）as (5).

$$p_{ij} = \prod_{k=1}^{j}(1-p_{ak})^{\frac{Tr}{Td}} * (1-(1-p_{a(j+1)})^{\frac{Tr}{Td}}) \tag{5}$$

p'_i indicates the probability of attackers having successfully acquired necessary resource R_i during time period T_r. We obtain p'_i *(*$1 \le i \le n$*)* as follows *(see* (6) ~ (8)).

$$p'_1 = p_1 * \left(1-p_{a1}\right)^{\frac{Tr}{Td}} \tag{6}$$

$$p'_2 = p_1 * (1-p_{a1})^{\frac{Tr}{Td}} * (1-p_{a2})^{\frac{Tr}{Td}} \tag{7}$$

$$p'_i = p_i * \prod_{k=1}^{i}(1-p_{ak})^{\frac{Tr}{Td}}, (3 \le i \le n) \tag{8}$$

p'_e indicates the probability that (1) attackers have successfully finished the attack and (2) all the necessary resources are not manipulated during time interval T_r. So, p'_e is expressed as formula (9).

$$p'_e = p_e * \prod_{k=1}^{n}(1-p_{ak})^{\frac{T_r}{T_d}} \tag{9}$$

We obtain the state transition matrix from the values of the probabilities p_{ij}、p'_i and p'_e. According to the Markov-Chain properties, the steady state probabilities of attackers staying in any state can be deduced. Then, we can derive the attack-success probability from formula (2).

We introduce γ - the attack-difficulty increment – to quantify security improvement offered by our DVM mechanism. Recall that p_{succ} and p'_{succ} are the attack-success probability before and after we apply the DVM mechanism. We refer to the discrepancy (i.e., p_{succ} - p'_{succ}) between p_{succ} and p'_{succ} as success probability reduction. We measure the attack-difficulty increment as the ratio between success probability reduction and probability p'_{succ}. Thus, we express the attach-difficulty increment as formula (10).

$$\gamma = \frac{p_{succ} - p'_{succ}}{p'_{succ}} = \frac{p_{succ}}{p'_{succ}} - 1 \tag{10}$$

Now we make use of a concrete example (i.e., n=3) to elaborate how to use our Markov-chain model to evaluate security improvement in terms of metric γ (see (10)). In this example, n being set to 3 means there are a total of three groups (i.e., R_1、$R_2$and R_3) of necessary resources for attackers.

We analyze the impacts of the dynamic-manipulation time interval (i.e., T_d) on the DVM security efficiency. We quantify the changes in user data attack-success probability and attack-difficulty increment as a function of T_d, which is set to 40, 45、50、55、60、65、70、100、200, and 100000. Note that when T_d equals to 100,000, the system exhibits a static behavior.

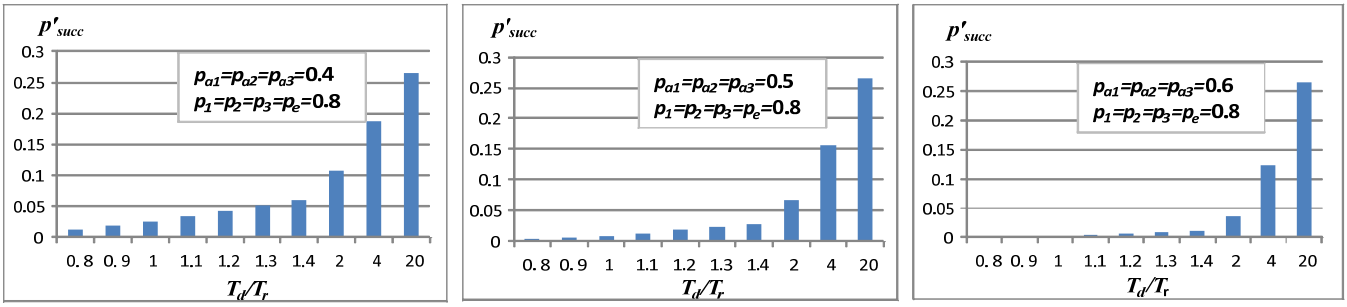
Table 2 shows the attack-success probability p'_{succ} as well as the attack-difficulty increment γ when T_d is increased from 40 to 100,000. In this numerical analysis, the necessary resource acquisition time interval (i.e., T_r) is set to a constant of 50; p_{a1}, p_{a2}, and p_{a3} are set to an identical value 0.5; p_1, p_2, p_3 and p_e are set to an identical value 0.8; the value of T_r is set to 50.

Table 2 Impact of T_d on attack-success probability p'_{succ} and attack-difficulty increment γ（T_r=50）.

T_d	40	50	55	60	65	70	100	200	1000	100000

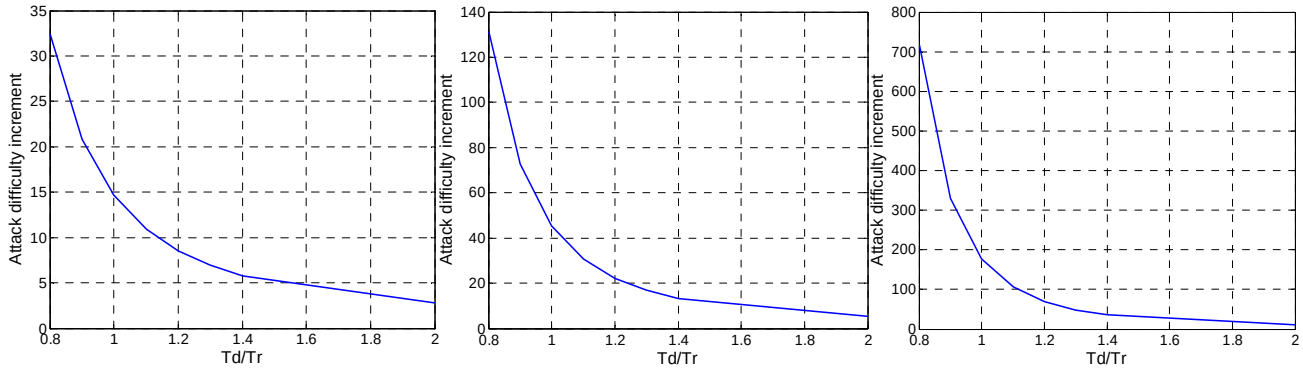
p'_{succ}	0.0031	0.0055	0.0088	0.013	0.018	0.023	0.029	0.067	0.157	0.265
γ	131.12	72.76	45.35	30.753	22.12	16.84	13.27	5.14	1.61	0.55

Figure B-2 shows the attack-success probability p'_{succ} as a function of ratio Tr/Td ; Figure B-3 shows the attack-difficulty increment γ as a function of ratio Tr/Td .In Figure B-2 and Figure B-3, we configure p_{a1}, p_{a2}, and p_{a3} with identical value and we configure this identical value to 0.4,0.5 and 0.6 to assess the impact of the probability value on security metrics attack-success probability and attack-difficulty increment. Table 2, Figure B-2 and Figure B-3 show that when T_d is smaller than or equal to $2T_r$ (i.e., $T_d \leq 2T_r$), attack success probability p'_{succ} is very low, and the attack-difficulty increment is very large. In the extreme case where T_d is smaller than T_r, (i.e., $T_d < T_r$), attack-success probability p'_{succ} is close to zero. This numerical analysis trend is consistent with our observations made in an empirical study. Thus, if the length of necessary resources changing period is less than the average time cost for attackers to obtain a set of necessary resources, attackers' attack-success probability p'_{succ} almost drops down to zero; if the length of necessary resources changing period is same as or slightly larger than the average time cost for attackers to obtain a set of necessary resources, attacks' attack-success probability is still maintain at a low level.



(a)$p_{a1}=p_{a2}=p_{a3}=0.4$ (b)$p_{a1}=p_{a2}=p_{a3}=0.5$ (c)$p_{a1}=p_{a2}=p_{a3}=0.6$

Figure B-2 Changes of attack-success probability with T_d/T_r.

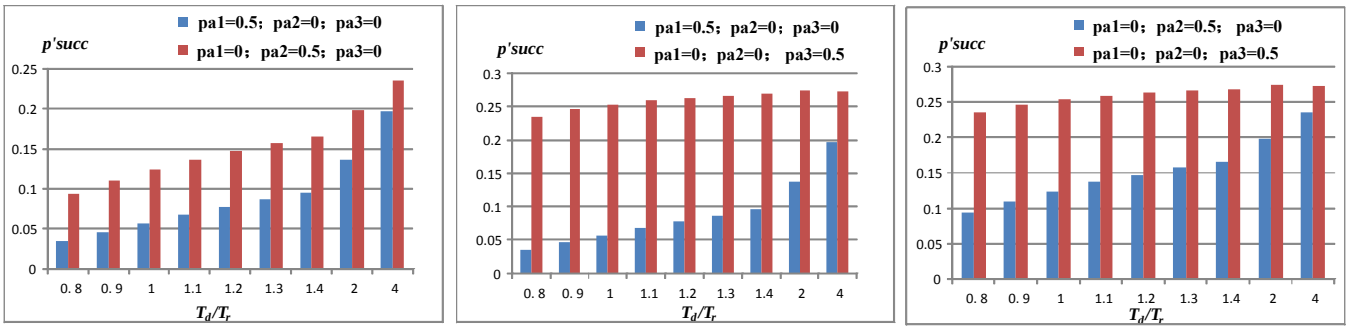


(a) *pa1=pa2=pa3*=0. 4 (b) *pa1=pa2=pa3*=0. 5 (c) *pa1=pa2=pa3*=0. 6

Figure B-3 Changes of attack-difficulty increment with T_d/T_r.

Figure B-2 and Figure B-3 intuitively show the relationships between the DVM mechanism's security efficiency and the necessary resources' change frequencies. In the DVM mechanism, a small T_d value leads to high security. Nevertheless, the small T_d value may introduce extra resource overhead. In our future study, we will focus on how to select an optimal T_d value for the DVM mechanism.

Now we analyze the impact of DVM on security efficiency when different necessary resource sets are dynamically manipulated. Again, we set p_1, p_2 , p_3 *and* p_e to 0.8 in this group of experiments. Figure B-4 and Figure B-5 respectively show attack-success probability and attack-difficult increment as a function of T_d/T_r when R_1, R_2 and R_3 are manipulated.

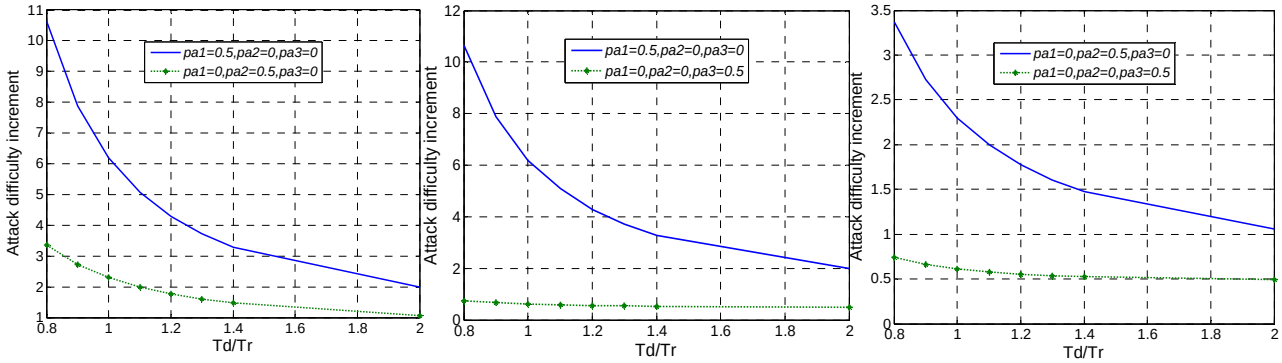


(a) Different attack success probability when only *R1* or *R2* is manipulated

(b) Different attack success probability when only *R1* or *R3* is manipulated

(c) Different attack success probability when only *R2* or *R3* is manipulated

Figure B-4 Impact of DVM on attack-success probability when different necessary resource sets are manipulated.



(a)Attack-difficulty increment when only *R1* or *R2* is manipulated

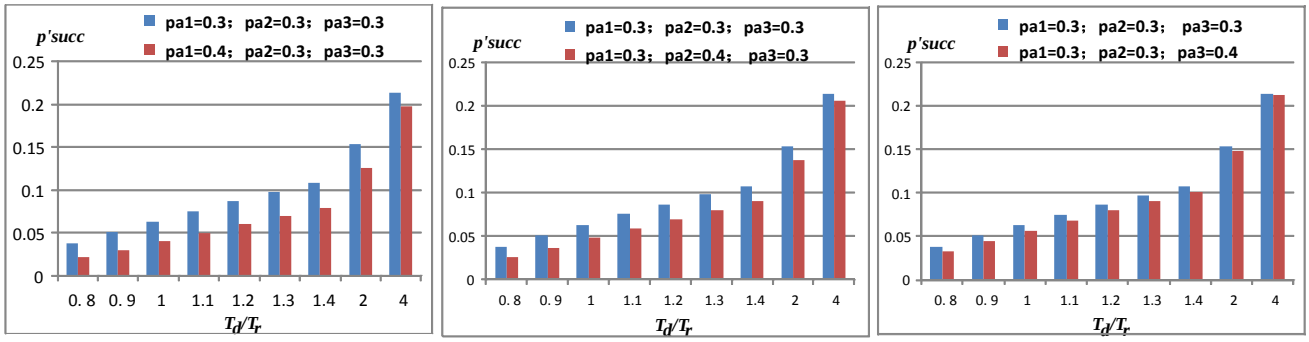
(b)Attack-difficulty increment when only *R1* or *R3* is manipulated

(c)Attack-difficulty increment when only *R2* or *R3* is manipulated

Figure B-5 Impact of DVM on attack-difficulty increment when different necessary resource sets are manipulated.

We observe from Figure B-4 and Figure B-5 that the security efficiency is larger when only R_1 is dynamically manipulated than when only R_2 or R_3 is dynamically manipulated. Similarly, dynamically manipulating R_2 better improves security than manipulating R_3. These results suggest that the security efficiency is higher when the necessary resources acquired in an earlier state are dynamically manipulated. Regardless of which state attackers are staying in, as long as necessary resource R_i that needs to be acquired earlier is dynamically manipulated, attackers must return back to state S_{i-1} and reacquire R_i as well as all the follow-up resources. From the perspective of attackers, a small value of i leads to a small attack-success probability or a large attack-difficulty increment.

Comparing Figure B-3 and Figure B-5, we can see that dynamically manipulating the three necessary resource sets (i.e., R_1, R_2, and R_3) offers more significant security improvement than simply manipulating one resource set.

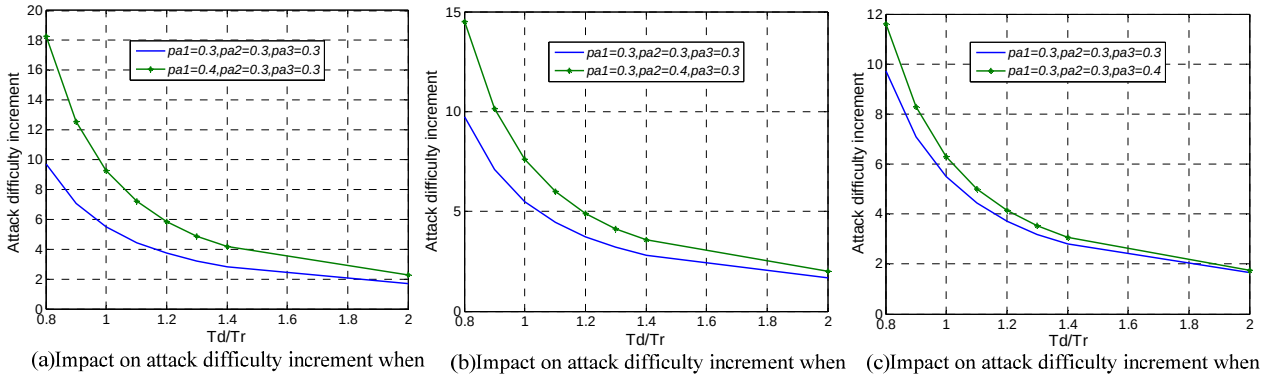


(a) Impact on attack success probability when the manipulation probability of *R1* is changed

(b) Impact on attack success probability when the manipulation probability of *R2* is changed

(c) Impact on attack success probability when the manipulation probability of *R3* is changed

Figure B-6 Changes of attack-success probability with dynamic-manipulation probabilities-1.



(a)Impact on attack difficulty increment when the manipulation probability of *R1* is changed

(b)Impact on attack difficulty increment when the manipulation probability of *R2* is changed

(c)Impact on attack difficulty increment when the manipulation probability of *R3* is changed

Figure B-7 Changes of attack-difficulty increment with dynamic-manipulation probabilities-1.

Figure B-6 and B-8 show the attack-success probability when the value of T_d/T_r is increased from 0.8 to 2; Figure B-7 and B-9 show attack-difficulty increment when the value of T_d/T_r is increased from 0.8 to 2.In this group of experiments, R_1, R_2 and R_3 are respectively dynamically manipulated with different probabilities.Figure11 and 12 reveal that regardless of changing which dynamic-manipulation probability, there are noticeable changes in security efficiency when T_d is less than or approximately equal to T_r. Comparing with R_2 and R_3, R_1's dynamic-manipulation probability has a larger impact on the defense efficiency. This result is consistent with that plotted in Figure B-5.

Figure B-8 and Figure14 show the trends of the security efficiency when the three dynamic-manipulation probabilities are all changed. Figure B-8 and Figure B-9 also illustrates when T_d is less than or approximately equal to T_r, the dynamic-manipulation probabilities have great impacts on the security efficiency.

We draw the following three conclusions from the aforementioned analysis. First, the DVM mechanism substantially improves overall system security by increasing attack difficulty. Second, two factors significantly affecting attack-success probability and attack difficulty include (1) the dynamic-manipulation time interval and (2) the occurrence probability. Last, but not least, a vital way of improving defense efficiency is through dynamic manipulations on necessary resources that must be acquired earlier.

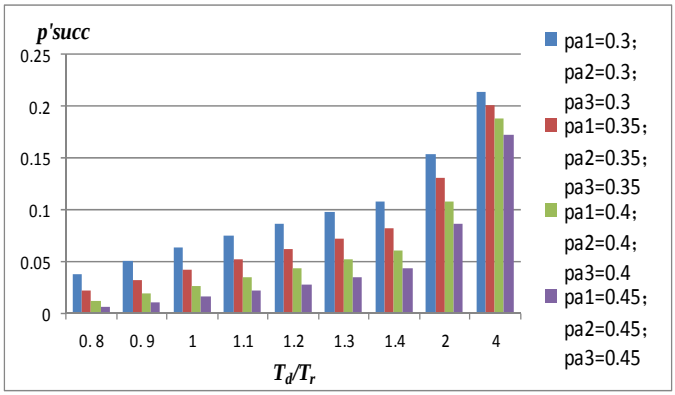


Figure B-8 Changes of attack-success probability with dynamic manipulation probabilities-2.

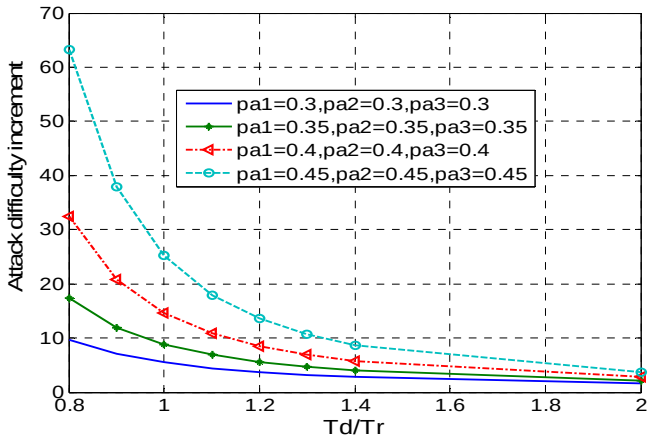


Figure B-9 Changes of attack-difficulty increment dynamic-manipulation probabilities-2.