Finite state machine for cloud forensic readiness as a service (CFRaaS) events

Victor R. Kebande1 | Kim-Kwang Raymond Choo2

1Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, Luleå, Sweden
2Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, Texas, USA

Correspondence
Victor R. Kebande, Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, Luleå, Sweden.
Email: victor.kebande@ltu.se

Abstract
The importance of demonstrating the correctness of forensic analysis tools and automated incident management tools reinforces the need for a finite state machine (FSM) engine that can generate automated forensic processes. Hence, in this paper, we present an event-based FSM representation for Cloud Forensic Readiness as a Service (CFRaaS), where we also show how the FSM’s predetermined states and transitions could be used to formulate an automated forensic process and generate a hypothesis for litigation purposes. Specifically, this proposition comprises a two-step level CFRaaS-FSM with possible transitions and states. This representation is useful because it can alert digital forensic investigators on how to deduce current and next state of attacks based on transitions and current states.

KEYWORDS
CFRaaS, cloud forensics, digital forensics, events, finite, level, machine, state

1 INTRODUCTION

The need to model discrete-event processes and control sequential logic partly contributes to the introduction of the Finite State Machine (FSM). Since then, FSM has been applied in a range of applications, for example, to solve supervisory problems in a more effective way by focusing on the formal language grammars, transitions that exist between states of conditioned machines. Generally, an FSM can traverse over predetermined sequence of states in order to achieve significant outcomes.

In this paper, we explore the utility of FSM in a digital forensic context. Specifically, while performing a digital investigation, an investigator may form a hypothesis by reconstructing or backtracking and tracing the occurrences. In order to form this hypothesis, through backtracking, the forensic analysis process should rely on a scientific theory, because this theory will explain how the expert conclusions are arrived at. Additionally, such an approach also results in objective data and clarification on how that data may be important. This also reinforces the importance of having a forensic-by-design process, as noted by Ab Rahman et al, and the fact that still informal analysis has been attributed to be inappropriate during complex investigations.

When a security incident occurs, the automated forensic analysis process can easily be used to trace the transitions that could have led to the state that transpired. In the context of FSM, it allows the investigator to be able to extract the FSM model, assess scenarios with the aim of formulating an automated forensic hypothesis that can be used for litigation...
purposes. In other words, using FSM for forensic analysis gives a sequential approach that allows an automated reconstruction of events during incident response, where the target system is modeled in a manner that allows the exploration of conditions and states under which evidence can be extracted.

Therefore, in this paper we demonstrate the potential of automating the CFRaaS event processes using FSM. This work extends the CFRaaS model presented by Kebande and Venter. Specifically, we present a conceptual two-step level FSM for CFRaaS, and explain how CFRaaS events can be automated using FSM. As a result, the contributions of this paper are as follows:

• Present a two-step event-based FSM representation for Cloud Forensic Readiness as a Service (CFRaaS).
• Demonstrate the correctness of forensic analysis tools and how FSM engine can generate automated processes in digital forensics.
• Show how FSM’s predetermined states and transitions could be used to formulate an automated forensic process and generate a hypothesis for litigation purposes

The remainder of the paper is organized as follows. In Section 2, a motivation is given, followed by Section 3, where we review the related literature and introduce the FSM for CFRaaS process events, respectively. In the last two sections, we present our discussion and concluding remarks.

2 | MOTIVATION

This study opines to the generic formal analysis of state machines from a digital investigation perspective. Basically, from a generic perspective, an event-based FSM is being applied to CFRaaS specifically from a cloud forensic perspective, by relying on the general aspects of state machines. For example, an FSM can widely be explored after being defined and based on the existing events/stages of CFRaaS, an FSM can be leveraged to determine or to give the probable scenarios that can lead to seamless analysis of a potential security incident based on some forensic hypothesis (Kebande et al., 2019; Adeyemi et al., 2017). That notwithstanding, an event-based FSM provides an approach of formally analyzing situations or circumstances through which forensic readiness process-dubbed events from a cloud forensic perspective, could in context be feasibly represented. This, is owing to the fact that, CFRaaS that previously has been described in Kebande and Venter, Kebande and Venter (2019b), and Kebande and Venter is composed of steps/processes/events that have a possibility of being formally analyzed as event-steps from a forensic examination perspective. Therefore, this paper employs the basic FSM approaches by highlighting generic states and transitions that could be used to formally analyze the CFRaaS events.

3 | BACKGROUND AND RELATED LITERATURE

3.1 | Finite state machine

A classic FSM is a sequential system that gives possible computations of a finite number of states and transitions among the states for some input and output conditions. Furthermore, an FSM is represented as an abstract automaton that works through a computer-aided mathematical model, where a user is able to define process (event) rules to parse the inputs. Each of the behaviors that is experienced by FSM begins with some requirement. This indicates that there always will exist an initial state in the FSM. In order for the FSM to be able to distinguish how the behaviors are activated, the aforementioned requirements need to have information in order to differentiate the behaviors. Therefore, two definitions for FSM and process communication in FSM are given as follows:

**Definition 1.** Finite State Machine (FSM): An FSM as is shown in Table 1 is defined as a sequential system that consist of 5 tuples, FSM = Q, , , , where as is shown in Table 1 Tuples, representations and descriptions are given:

**Definition 2.** Process communication in FSM: In an FSM, a process communication (PC-FSM) shown in Table 2, is being represented as an instance that consist of a set of machines and a path between the machines that allows effective communications based on states, transitions and actions.
TABLE 1  FSM 5-tuple representation

| No | Tuple | Representation | Description |
|----|-------|---------------|-------------|
| 1  | $Q = \{q_1, q_2, \ldots, q_n\}$ | Finite nonempty set of states | FSM can only be in one definite state, $q_i$, where $i = 1, 2, \ldots, n$ |
| 2  | $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_m\}$ | Set of input symbols | FSM can only receive certain input, $\sigma_j$, $j = 1, 2, \ldots, m$ |
| 3  | $\delta : (Q \times \Sigma \rightarrow Q)$ | A state transition function | When FSM receives some input, it changes from definite state |
| 4  | $q_0 \in Q$ | Is the initial state | FSM starts to receive input from initial state |
| 5  | $F \subseteq Q$ | Set of end states | FSM stops to receive any input |

TABLE 2  Process communication

| Process communication | Description |
|-----------------------|-------------|
| 1 $K = (S, P)$        | $S$ is a set of FSMs, $S = \{FSM_1, FSM_2, \ldots, FSM_n\}$ |
| 2 $P = \{P_{ij} | i < r, \text{and } i \neq j\}$ | Path between FSM$_i$ and FSM$_n$ |

As a result, the author represents PC-FSM as follows:

During the process communication, each relationship formed holds some relevance with each state. In other words, in each $K = (S, P)$, a relationship will describe every instance during communication. Also, by assuming a given path, PC-FSM facilitates the actions from one state to the other. Similarly, Ikuesan and Venter$^{16}$ explained that the actions from one state to the other can be encoded to allow the sequence of actions to be computed. This could also be applicable in the aforementioned proposition.

3.2  Foundation of cloud forensic readiness

The foundations of digital forensic readiness on cloud computing technologies has been necessitated by increased usage of cloud computing resources and the rapid rise of potential security incidents.$^{17,18}$ The technical factors like cloud infrastructure, cloud architecture, forensic technology affect how cloud forensic readiness aspects should be conducted. There still exist other researches that have depicted how services can be formalized in a cloud forensic readiness approach, for example, leveraging SLAs formalization based on cloud forensic reference architecture,$^{19}$ forensic readiness in the cloud through forensics log collection,$^{20}$ decision making approaches using cloud forensic readiness approaches.$^{21}$ Other pertinent research on cloud forensics has mainly focused on presenting holistic ontology-driven perspectives on cloud forensic aspects where readiness is employed as a service based on the simplicity of concepts, relationships and semantics of the potential evidence that can be used for investigation purposes.$^{22,23}$ Notably, a taxonomy of cloud forensic solutions has shown that cloud forensics has a focus on the incidents that emanates from the cloud infrastructures and activities that are concerned in criminal investigations and litigations. Basically, cloud forensics has become increasingly important in the recent past.$^{24}$ While these development and challenges presents cloud forensics to be more complex than traditional forensics due to its architectural complexity, it is important to explore some of the challenges that hampers successful forensic investigations.

3.3  CFRaaS events

Based on the dynamisms of the FSM, the CFRaaS events can be represented, where actions can be formally used to represent automated processes of forensic analysis. The CFRaaS’s main aim is to assist forensic experts to achieve timely analysis and as a result the representation of CFRaaS events has been given based on the interaction of processes. The aim is to be able to find most or all possible events that can be used to extract forensically sound evidence with respect to FSM. The construction of CFRaaS-FSM events is useful because it explores all possible computations with the aim of automating forensic analysis process by speeding up the formal analytical processes,$^5$ as is shown in Figure 1.
Figure 1 shows a high-level classification of CFRaaS events and processes, where a high-level CFRaaS consist of four distinct processes illustrated as layers: provider layer, virtualization layer, Digital Forensic Readiness (DFR) layer and Incident Response Procedure (IRP) layer, respectively.22,23

4 | FSM FOR CFRAAS PROCESS EVENTS

The concept of CFRaaS events is aimed at formally providing an automated forensic analysis or incident management process based on states and transitions among the events. The author opines to the fundamental representation for the FSM, and based on this, the FSM-CFRaaS event has been represented on two levels namely: FSM-CFRaaS-Level 1 and FSM-CFRaaS-Level 2, respectively, which are stages of formally representing the automated processes.

4.1 | Formal description

A general state machine (Turing machine) is formally represented as a mathematical model that possess an infinite size of cells that allow a given number of inputs, where each cell consist of symbols 0 or 1, with states and actions.25-27 Consequently, the description is also based on the above-mentioned definition (Section 3), where, for example, the transitions in an FSM that is shown in Figure 2 can be represented based on the given states X, Y, and Z, respectively.

A formal FSM, for example in Figure 2 can generate a sequence of transitions that can be computed, where this computation is able to start at any given state. For example, Figure 1 can produce the following transitions: \([X \rightarrow 1 \rightarrow Y]\), \([Y \rightarrow 1 \rightarrow Y]\), \([X \rightarrow 1 \rightarrow Z]\), & \([Z \rightarrow 1 \rightarrow Z]\), respectively, which can be decoded deterministically given that the next state can uniquely be determined based on a single input (event).26,28
Formally, in a forensic investigation perspective, an FSM can be applied based on the assumptions that a forensic hypothesis holds. A forensic hypothesis in this context is based on processes that are generated, for example with CFRaaS, whose end result is the extraction of useful potential evidence. Notably, based on these processes, an event-based FSM in the context of this paper is used to theorize or formalize the assumptions that the forensic hypothesis holds—this is achieved by providing key investigative assumptions that can lead to the discovery of digital evidence in an automated fashion through all possible outcomes that an FSM can represent.

4.2 | Event-based CFRaaS-FSM level-1

Level-1 of the CFRaaS-FSM allows a forensic investigator to initialize the CFRaaS process in order to formalize the events when the process is triggered as shown in Figure 2. From the theoretical perspective, formal analysis is achieved by generating CFRaaS-FSM process and then determining and executing all the possible scenarios that are linked to security incident by observing the respective states and transitions. Figure 3 illustrates the possible scenario for Event-based CFRaaS-FSM-level 1, where events are formalized based on the executed processes with $Q = \{\text{start, stop}\}$, $\Sigma = \{0, 1\}$, $q_0 = \text{start}$, $F = \text{stop}$, respectively.

From level 1, CFRaaS, the incoming arrow depicts the CFRaaS process initial state, while the transitions are depicted by the arrow originating from the start state. The conditions and actions of CFRaaS events are labeled along the arrows as is shown in Table 3.

4.3 | Event-based CFRaaS-FSM level-2

Level-2 of the CFRaaS-FSM allows events to be parsed based on some inputs. For example, a number of high-level events are highlighted that basically in the abstract model are represented as high-level subprocesses. From this level, the paths around the CFRaaS are defined in such a manner that they are able to accept the rules through which the events that are generated are triggered on. As shown, the FSM considers 12 high-level processes across the four high-level CFRaaS processes (provider $P_l$, virtualization $V_l$, DFR DFR_l and Incident Response Procedure IRP_l), which are later decomposed into event states as follows: service provision ($S_p$), Monitoring ($VM_t$), VM provisioning ($VM_p$), Instance management ($I_m$), Evidence collection ($E_c$), Pre-incident Analysis ($PL_a$). Incident Identification ($I_i$), Event

![Event-based CFRaaS-FSM-Level 1](image)

**Figure 3** Event-based CFRaaS-FSM-Level 1

| Table 3 CFRaaS-FSM-Level 1 transitions |
|---------------------------------------|
| Start | Running CFRaaS process | 1 |
| Stop  | CFRaaS process halt    | 0 |
Reconstruction (Er), Dynamic Reporting (Dr), Initialization process (Iinitp), Acquisitive process (Aqrip) and Investigative (Iirp), respectively.

Based on the events that are generated from Figure 4, the authors highlight a two-step process FSM for the tuple that are based on Layer 1 and Layer 2 of the CFRaaS as is shown in Figure 5, where the number of possible states and transitions are represented as is sown in Equation (1):

\[ Q = \{P, S_p, VM_t, VM_p, I_m\} \]

\[ \sum = \{0, 1\} \]

\[ q_0 = \{P\} \]

\[ F = \{S_p, I_m\} \]

where the \( \delta : \{Q \times \sum \rightarrow Q\} \) for the 2-step level FSM shown in Figure 5 is given by the transition table that show all the possible computations for the events that are generated at the 2-step phase and this is shown in Table 4 where \( X_1, X_2, \) and \( X_3 \) are the CFRaaS FSM states and \( X_i \) is an instance of CF. The 2-step level CFRaaS-FSM has five independent cycles through which the FSM is able to move a number of times so that it can arrive at the accepting state. For illustration purposes, the authors utilize cycle \([P, S_p]\) and \([P, V]\), for example, taking two distinct cycles \([P\rightarrow V\rightarrow P]\) and \([V\rightarrow M_t\rightarrow V]\). This indicates that the 2-step level CFRaaS-FSM can accept a string based on the following cycle representations: \([P\rightarrow V]\rightarrow a, [V\rightarrow P]\rightarrow b, [V\rightarrow M_t]\rightarrow b, [M_t\rightarrow V]\rightarrow b\). Based on these independent two cycles this 2-step CFRaaS-FSM can accept a regular expression \((ab + bb)\) with respective strings. In this context, matching a string from a given regular expression \(R\), follows the fundamental process where a regular expression, \(R\), takes the length of an expression \(m\) and the string \(Q\) with its specific length \(l\). It is possible for \(m \leq l\). The context of this study is inclined to how the specific CFRaaS events are able to adopt the same cycles during an investigation approach until they are truncated. For example, the regular expression \((ab + bb)\) would accept a number of strings like \(ab, bb\) through which, it is able to translate and fit into the 2-step level CFRaaS FSM. Important to note is the fact that the two cycles have been used for illustration purposes owing to the fact that our proposition represents a generic view of CFRaaS-FSM. While also it is possible to take loop transitions within the states, the decision on which transition is to be taken depends on the task that the CFRaaS-FSM needs to read.

The author posits that level-1 and level-2 of the CFRaaS-event-based FSM gives an approach of formalizing and automating tool development process, whereby each event that has been generated is able to give an outline how it is related to a particular security incident. Therefore, a calculation of the number of transitions that are generated in the 2-step-level event based CFRaaS-FSM in Table 4, can be calculated as is shown in Equation (2):

\[ \sum_{i=1}^{n} \cdot \sum_{j=1}^{n} (CF)_{ij} \]

where \( CF \) represents the CFRaaS-FSM and \((CF)_{ij}\) represents the total transitions over 0 and 1 as has previously been portrayed in Table 4. Based on the \( \delta : \{Q \times \sum \rightarrow 2Q\} \), the set of all possible transitions has been shown in Table 4 is

---

**Figure 4** Level-2, CFRaaS events
Table 4: 2-step-level event-based CFRaaS FSM transitions

| CF  | $X_1$ | $X_2$ | $X_3$ |
|-----|-------|-------|-------|
| $Q$ | $P$   | $[P]$ | $[S_p, V, P]$ |
| $X_2$ | $S_p$ | $[P]$ | $-$ |
| $X_3$ | $V$   | $[P, V]$ | $[M_t, V]$ |
| $X_4$ | $M_t$ | $[V]$ | $[VM_p]$ |
| $X_5$ | $VM_p$ | $[M_t]$ | $[I_m]$ |
| $X_6$ | $I_m$ | $[VM_p, I_m]$ | $[I_m]$ |

Table 5: A 2-step-level event-based CFRaaS FSM, events, states, and possible transitions

| 2-step layers | Events          | Current state | Possible transitions |
|---------------|-----------------|---------------|----------------------|
| 1             | Provider        | $[S_p]$       | $[S_p \rightarrow S_p]$ |
| 2             | Virtualization  | $[M_t, VM_p]$ | $M_t \rightarrow VM_p$, $M_t \rightarrow M_t$, $VM_p \rightarrow M_t$, $M_t \rightarrow I_m$, $I_m \rightarrow M_t$, $I_m \rightarrow I_m$ |
|               | Monitoring      | $[M_t, VM_p, I_m]$ | $M_t \rightarrow VM_p$, $M_t \rightarrow M_t$, $VM_p \rightarrow M_t$, $M_t \rightarrow I_m$, $I_m \rightarrow M_t$, $I_m \rightarrow I_m$ |
|               | VM provisioning | $[M_t, VM_p, I_m]$ | $M_t \rightarrow VM_p$, $M_t \rightarrow M_t$, $VM_p \rightarrow M_t$, $M_t \rightarrow I_m$, $I_m \rightarrow M_t$, $I_m \rightarrow I_m$ |
|               | Instance management | $[M_t, VM_p, I_m]$ | $M_t \rightarrow VM_p$, $M_t \rightarrow M_t$, $VM_p \rightarrow M_t$, $M_t \rightarrow I_m$, $I_m \rightarrow M_t$, $I_m \rightarrow I_m$ |

represented as follows (See Table 5): Important to note is the fact that the study is presented in a generic perspective as an opinion and as a result at this level it is not explicitly inclined to deterministic FSM or non-deterministic FSM.

Furthermore, given that security incidents can occur at any given time and from any given source, the event-driven approach collates the events in a more realistic approach, monitors their relationships, gives a description of each event in order to increase the degree of incident detection. The ultimate goal of this opinion paper is to illustrate how the use of FSM related approach can offer simplicity during incident detection.
5 | DISCUSSIONS

Digital forensics is still evolving as a discipline and it is important to automate the digital forensic tools, that can allow human to be able to highlight how the formal description of the forensic hypothesis, incidents, and the digital ecosystem correlates. Therefore, the author takes a step to show the need for developing a generic event-based FSM for CFRaaS, which in this context, is a formal approach that can be applicable when developing forensic analysis tools. The generic approach that has been proposed in this paper relies on an FSM to execute its tasks (states and transitions) which may be useful in reconstructing events, identifying investigative scenarios and the relationship that may exist between the states and the transitions.

The core challenge that exist among the forensic tools according to Gladyshev and Patel\textsuperscript{2)} is the difficulty in proving the correctness of forensic analysis techniques. It is therefore important to note that formalizing the CFRaaS event states, generates the systems’ knowledge which makes it easy to prove the correctness of the states and transitions. For example, while a four-step high-level CFRaaS with 12 events: (provider ($P_1$), virtualization ($V_1$), DFR (DFR$_1$) and Incident Response Procedure (IRP$_1$)), which are later decomposed into event states as follows: Service provision ($S_p$), Monitoring ($M_t$), VM provisioning ($VM_p$), Instance management ($I_m$), Evidence collection ($E_c$), Pre-incident Analysis ($PL_a$), Incident Identification ($II_p$), Event Reconstruction ($E_r$), Dynamic Reporting ($DR$), Initialization ($In_{irp}$), Acquisitive ($A_{irp}$) and Investigative ($I_{irp}$), respectively, could produce numerous states and transitions, the author has mainly focused on highlighting the outcome of a two-step process (event).

The two-step process that has been shown in Figure 5 and transition Table 4 shows that the interpretation could be essential in sharing the approaches of creating a finite machine with a higher degree of correctness and proof of forensic approaches. Additionally, the proposed approach shows fundamentally how a variation of events could yield some output which under some situation or condition may also be deterministic or indeterminate.

In the two-step level approach, the authors represented two high-level processes (Provider and virtualization) with $Q = \{P, S_p, V, M_t, VM_p, I_m\}$ events and $q_0 = \{P\}$ as the starting state and two final states $F = \{S_p, I_m\}$. The processes are executed when the transitions changes based on some given inputs that are denoted as $\Sigma = \{0, 1\}$. That notwithstanding, the interpretation could in the long run provide strong evidential guarantee with a higher degree of acceptance on the forensic community if incorporated in tool development. Therefore, based on the opinions that have been put across in this paper, the author ascertains as follows:

1. It is possible to generate conditioned forensic tools that can easily be used to detect potential faults that hinders incidental detection by way of providing proofs and correctness.
2. Based on how the events in CFRaaS relate, it is possible to automatically create a cohesive relationship among automated processes during forensic analysis, because this can easily give a formal description of each event (process).
3. From the 2-step CFRaaS FSM, it is important to note that, based on some inputs, back-tracing events by way of constructing event transitions can accelerate detecting with a degree of accuracy.

Consequently, the CFRaaS-FSM that has been described in this paper has been presented from a generic perspective because the objective of the study is focused on giving an overview/abstract representation on how cloud forensic processes (events) can be represented from the perspective of a state machine. Also, the intention of CFRaaS-FSM is to be able to conceptually show how integrating these events to FSM could provide an easy way of conducting digital forensic processes in the cloud environment from the perspective of a FSM. As a result, the events that have been used in the CFRaaS-FSM representation simply portrays some of the activities that emanates from the CFRaaS model that was initially suggested by Kebande et al,\textsuperscript{22,23} Kebande and Venter.\textsuperscript{12} On the same note, the FSM’s representation has been leveraged to explicitly depict from a generic perspective, how a cloud forensic investigation process could be modeled based on initially created steps/subprocesses. The concluding remarks are given in the next section.

6 | CONCLUDING REMARKS AND FUTURE WORK

A technique for automating CFRaaS process events using FSM was presented in this opinion paper. Based on the propositions that have been suggested in this opinion paper, the approach can from a generic stand point be employed in automating a cloud forensic readiness tool. We believe that such an approach can facilitate digital forensic investigation
Opinion of FSM for CFRaaS events

FSM for CFRaaS events has been conceptualized as a very important step toward forensic analysis tool automation and digital evidence formalization. Basically, this approach has been explored, owing to the need for illustrating how conditioned machines in digital forensics can use determinate and predeterminate states to detect probable and possible occurrences. This is useful because it increases the chances of causality and admissibility by way of generating accurate hypothesis and expert conclusions.

while leveraging formal descriptors based on, for example, CFRaaS steps (potential evidence, and core hypothesis based on the existing forensic scenario).

Future research will include embedding the automated processes into tool development processes. Additionally, by formalizing each step or process of the CFRaaS, the study aims to identify key features and attributes of CFRaaS that can be used from the perspective of digital forensic investigative hypothesis.

Since our suggested CFRaaS-FSM has only been presented from a generic perspective, in the future one can consider extending the CFRaaS-FSM to accommodate formal heuristics, say formalizing CFRaaS-FSM using verifiable notations in order to prove the correctness and accuracy of the steps that have been undertaken.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their positive and constructive feedback. In addition, the authors would like to acknowledge that the opinion, findings, and conclusions expressed in this article are purely of the authors.

CONFLICT OF INTEREST

The authors declare no potential conflict of interest.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

ORCID

Victor R. Kebande https://orcid.org/0000-0003-4071-4596
Kim-Kwang Raymond Choo https://orcid.org/0000-0001-9208-5336

REFERENCES

1. Ramadge PJ, Wonham WM. Supervisory control of a class of discrete event processes. *SIAM J Control Optim*. 1987;25(1):206-230.
2. Gladyshev P, Patel A. Finite state machine approach to digital event reconstruction. *Digit Invest*. 2004;1(2):130-149.
3. Ab Rahman N.H., Glisson W.B., Yang Y., Choo K.K.R. Forensic-by-design framework for cyber-physical cloud systems. *IEEE Cloud Comput*. 2016;3(1):50-59.
4. Chen L, Wang G. General finite state machine reasoning method for digital forensics. *Mobile Multimedia/Image Processing, Security, and Applications 2008*. Vol 6982. Bellingham, Washington: International Society for Optics and Photonics; 2008:69820I.
5. Gladyshev P, Patel A. Finite state machine analysis of a blackmail investigation. *Int J Digit Evidence*. 2005;4(1):1-13.
6. Kebande VR, Venter HS. Novel digital forensic readiness technique in the cloud environment. *Aust J Forensic Sci*. 2018;50(5):552-591.
7. Karie NM, Kebande VR, Venter HS, Choo KKR. On the importance of standardising the process of generating digital forensic reports. *Forensic Science International: Reports*. 2019;1:100008.
8. Adeyemi, I. R., Abd Razak, S., Salleh, M., & Venter, H. S. (Indonesia: Indonesian Society for Knowledge and Human Development; 2017). *Leveraging Human Thinking Style for User Attribution in Digital Forensic Process*.
9. Adeyemi IR, Abd Razak S, Salleh M. Understanding online behavior: exploring the probability of online personality trait using supervised machine-learning approach. *Front ICT*. 2016;3:8.
10. Mohlala M, Ikuesan AR, Venter HS. User attribution based on keystroke dynamics in digital forensic readiness process. *2017 IEEE Conference on Application, Information and Network Security (AINS)*. Miri, Malaysia: IEEE; 2017:124-129.
11. Lagrasse M, Singh A, Munkhondya H, Ikuesan A, Venter H. Digital forensic readiness framework for software-defined networks using a trigger-based collection mechanism. *Proceedings of the 15th International Conference on Cyber Warfare and Security, ICCWS*; Norfolk, Virginia: ACPI; 2020:296-305.
12. Kebande VR, Venter HS. A comparative analysis of digital forensic readiness models using CFRaaS as a baseline. *Wiley Interdiscip Rev Forensic Sci*. 2019;1(6):e1350.

13. Kebande VR, Venter HS. CFRaaS: Architectural design of a Cloud Forensic Readiness as-a-Service Model using NMB solution as a forensic agent. *African Journal of Science, Technology, Innovation and Development*. 2019b:1-21.

14. Wu S. The implement of animation state transitions in interactive scenes based on graphic finite state machine. *2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics*. Vol. 2. Nanchang, China: IEEE; 2012:242-245.

15. Lin X. Multi-behaviors finite state machine. *2009 IEEE Youth Conference on Information, Computing and Telecommunication*. Beijing, China: IEEE; 2009:201-203.

16. Ikuesan AR, Venter HS. Digital behavioral-fingerprint for user attribution in digital forensics: are we there yet? *Digit Investig*. 2019;30:73-89.

17. Alenezi, A., Zulkipli, N. H. N., Atlam, H. F., Walters, R. J., & Wills, G. B. (New York: ACM; 2017a). The impact of cloud forensic readiness on security. In *CLOSER* (pp. 511-517).

18. Alenezi A, Hussein RK, Walters RJ, Wills GB. A framework for cloud forensic readiness in organizations. *2017 5th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*. San Francisco, CA: IEEE; 2017b:199-204.

19. De Marco L, Abdalla S, Ferrucci F, Kechadi MT. Formalization of slas for cloud forensic readiness. *Proc. ICCSM Conference*; Reading: ACI; 2014:42-50.

20. Trenwth, P. M., & Venter, H. S. (2013). Digital forensic readiness in the cloud. In *2013 Information Security for South Africa* (pp. 1-5). Johannesburg, South Africa: IEEE.

21. Simou, S., Troumpis, I., Kalloniatis, C., Kavroudakis, D., & Gritzalis, S. (2018). A decision-making approach for improving organizations’ cloud forensic readiness. In *International conference on trust and privacy in digital business* (pp. 150-164). Springer, Cham.

22. Kebande VR, Baror SO, Parizi RM, Choo KKR, Venter HS. Mapping digital forensic application requirement specification to an international standard. *Forensic Sci Int Rep*. 2020a;2:100137.

23. Kebande VR, Karie NM, Ikuesan RA, Venter HS. Ontology-driven perspective of CFRaaS. *Wiley Interdiscip Rev Forensic Sci*. 2020b;2(5):e1372.

24. Manral B, Somani G, Choo KKR, Conti M, Gaur MS. A systematic survey on cloud forensics challenges, solutions, and future directions. *ACM Comput Surveys (CSUR)*. 2019;52(6):1-38.

25. Barker-Plummer, D. (1995). Stanford: Stanford University. *Turing Machines*.

26. James J, Gladyshev P, Abdullah MT, Zhu Y. Analysis of evidence using formal event reconstruction. *International Conference on Digital Forensics and Cyber Crime*. Berlin, Heidelberg: Springer; 2009:85-98.

27. Rogozhin Y. Small universal Turing machines. *Theor Comput Sci*. 1996;168(2):215-240.

28. Parker PM (Ed.). *Finite State Machine*; 2009. Webster’s Online Dictionary. Retrieved February 11, 2009. http://www.webstersonlinedictionary.org/fi/finite+state+machine.html.

29. Ikuesan AR, Venter HS. Digital forensic readiness framework based on behavioral-biometrics for user attribution. *2017 IEEE Conference on Application, Information and Network Security (AINS)*. IEEE; 2017:54-59.

---

**How to cite this article:** Kebande VR, Choo K-KR. Finite state machine for cloud forensic readiness as a service (CFRaaS) events. *Security and Privacy*. 2021;e182. [https://doi.org/10.1002/spy2.182](https://doi.org/10.1002/spy2.182)