Claims

1. **A system** for detecting synthetic or simulated keystroke activity on a computing device, the system comprising:
– **a keystroke monitoring module** configured to record keyboard input events and timing data from a user's computing session;
– **a behavioral analysis module** operatively coupled to the keystroke monitoring module, the behavioral analysis module programmed to analyze timing intervals and patterns of the recorded keyboard input events to determine a plausibility score indicative of human typing randomness versus automated input regularity;
– **an application context module** configured to track a foreground application and active user interface element receiving input during the computing session, and to correlate the recorded keyboard input events with the foreground application's context, wherein the application context module generates a context mismatch signal upon detecting keystroke events that are inconsistent with the active application state or UI element (such as typing occurring in a non-text field or an inactive window);
– **a screen content analysis engine** configured to capture screen output data from the computing device (via screenshots or video frames) and to detect on-screen changes corresponding to the keyboard input events, the screen content analysis engine producing an output inconsistency signal if a predefined volume of keystrokes occur without expected corresponding changes in the screen content (including detecting lack of new text, cursor movement, or UI updates);
– **a facial recognition module** interfaced with a camera to detect a presence of an authorized user during the computing session, the facial recognition module verifying user identity and generating a user-absence signal when no face or an unrecognized face is detected during periods of keyboard activity;
– **an anomaly detection aggregator** configured to receive inputs from the behavioral analysis module, the application context module, the screen content analysis engine, and the facial recognition module, the anomaly detection aggregator applying one or more rules or algorithms to combine said inputs into an overall determination of whether the keyboard input events are likely synthetic or inconsistent with genuine user activity; and
– **an alert and logging module** responsive to the anomaly detection aggregator, the alert and logging module being configured to generate an alert notification or log entry when the overall determination indicates synthetic keystroke activity beyond a defined threshold, wherein the alert notification comprises information identifying the type of inconsistency detected among timing, context, screen content, and user presence.

2. The system of claim 1, wherein the **behavioral analysis module** computes metrics including at least one of: average and variance of inter-keystroke intervals, frequency of repeated keystroke patterns, typing burst rates (keystrokes per unit time), and a randomness entropy measure of keystroke sequences, and wherein the behavioral analysis module flags an anomaly when one or more of said metrics exceed predefined human physiological thresholds or deviate from a historical profile of the specific user.

3. The system of claim 1, wherein the **application context module** further comprises an interface to an operating system's accessibility or UI automation framework to determine a type of active UI control, and the module suppresses the context mismatch signal for predefined benign scenarios including global hotkey combinations or operating system command shortcuts, such that normal system operations do not trigger false positive context mismatch alerts.

4. The system of claim 1, wherein the **screen content analysis engine** utilizes optical character recognition (OCR) to read text from captured screen images and compare it to recently logged keyboard inputs, raising the output inconsistency signal if, after a series of character-input keystrokes, the OCR-detected text on the screen fails to reflect the majority of those inputs, thereby detecting "invisible" or ineffective keystroke events.

5. The system of claim 1, wherein the **screen content analysis engine** is further configured to detect movement of a text cursor or focus indicator in the screen output data, and to determine when navigation keystrokes (arrow keys, tab, backspace, etc.) result in corresponding moves of said cursor or focus; the engine contributes to the output inconsistency signal upon detecting a prolonged discrepancy (beyond a preset duration) between navigation keystrokes and any visible cursor movement or focus change.

6. The system of claim 1, wherein the **facial recognition module** includes a liveness detection sub-module requiring one or more spontaneous user actions (including blinking, smiling, or head movements) to verify that the detected face is live and not a static image, thereby ensuring that the presence of the correct user is genuine during active periods of the session.

7. The system of claim 1, wherein the **facial recognition module** is configured to continuously or periodically authenticate the user's identity during the session and to generate an immediate security alert in addition to the user-absence signal if a different person is detected using the terminal while keyboard activity is ongoing, thereby providing an additional layer of access control by tying input events to the authenticated user's identity.

8. The system of claim 1, wherein the **anomaly detection aggregator** implements a weighted scoring algorithm that assigns weights to the signals from the behavioral analysis, application context, screen content, and facial recognition modules, and computes a composite risk score, and wherein the alert and logging module triggers an alert when the composite risk score exceeds a configurable threshold, thereby allowing tuning of the system's sensitivity to synthetic activity.

9. The system of claim 8, wherein the weights or threshold in the weighted scoring algorithm are adjustable based on an **administrative override or machine learning model**, such that the system can be calibrated to a specific environment – for example, reducing sensitivity during periods of known automated maintenance (thus preventing alerts) or increasing sensitivity for high-security modes where even subtle anomalies are reported.

10. The system of claim 1, wherein the alert and logging module comprises an administrative dashboard GUI that presents **correlated anomaly data** with timestamps and categories of detected anomalies, and further allows an administrator to review supporting evidence including portions of keystroke logs, screenshots of the screen at the time of anomaly, and snapshots of the user's face (if available), to facilitate rapid investigation and confirmation of each alert.

11. The system of claim 1, wherein the alert and logging module is further configured to execute a **mitigative action** in response to a detected anomaly, the mitigative action selected from the group consisting of: temporarily pausing or buffering further keystroke input to the active application, locking the user's session and requiring re-authentication, notifying a remote security server or supervisor via network message, and marking the user's account for heightened scrutiny, thereby actively preventing potential malicious consequences of synthetic input in addition to logging it.

12. **A method** for detecting synthetic keystroke activity in a user session, the method comprising:
(a) **Recording**, via a monitoring process, a sequence of keystroke events including their timestamps and the identity of keys pressed during a user session on a computing device;
(b) **Determining an active application context** by identifying at intervals which application and interface element is currently focused to receive input on the computing device;
(c) **Capturing screen images** of the display of the computing device periodically and/or in response to predetermined triggers during the session;
(d) **Detecting user presence** by capturing images of the user (through a camera) and attempting facial identification against an expected user profile during the session;
(e) **Analyzing the keystroke events' timing and patterns** to compute one or more behavioral metrics, and flagging a timing anomaly if said metrics fall outside ranges associated with human typing behavior;
(f) **Comparing keystroke events to application context** to determine if the keystrokes are appropriate for the active application or UI element, and flagging a context anomaly upon detecting keystrokes that would have no valid effect or meaning in the identified context;
(g) **Analyzing the captured screen images** to identify changes corresponding to the keystroke events, including using image analysis to detect new text or UI updates, and flagging an output anomaly if a sequence of keystrokes yields no commensurate change in the screen images;
(h) **Verifying the user's presence** by analyzing the captured user images for the expected user's face, and flagging a presence anomaly if the expected face is not continuously present during periods of intense keystroke activity (or if a face is not detected at all);
(i) **Aggregating any anomalies** flagged in steps (e), (f), (g), and (h) to determine, based on predefined rules or an algorithm, whether the combination of anomalies meets criteria indicative of synthetic or simulated keystroke activity; and
(j) **Generating an alert or log entry** if the criteria are met, the alert indicating that synthetic keystroke activity is suspected and providing information about which anomalies were detected, wherein if no criteria are met the method continues monitoring the user session without alert until an anomaly is detected or the session ends.

13. The method of claim 12, wherein step (e) comprises calculating an average keystroke rate over a rolling time window and comparing it to a maximum human keystroke rate threshold, and further comprises identifying patterns of equal time intervals between successive keystrokes using statistical tests, such that the timing anomaly is flagged when either sustained typing speed exceeds the threshold or when the distribution of inter-keystroke intervals shows a statistically significant uniformity indicative of programmatic input.

14. The method of claim 12, wherein step (f) comprises: checking the type of the active UI element (window or control) at the time of each keystroke, determining whether the keystroke is a character input, navigation command, or shortcut, and for each keystroke, verifying that it aligns with an expected action in the UI element (for example, character inputs only when a text field is focused, arrow keys only when a scrollable or navigable element is in focus); the context anomaly is flagged when keystrokes occur that would be ignored or are irrelevant given the UI element's state, indicating a likely automated sequence not actually intended for the open interface.

15. The method of claim 12, wherein step (g) further includes computing a **screen change ratio** defined as the amount of pixel or content change detected in the screen images per number of keystrokes in a time period, and comparing this ratio to a minimum expected change ratio, such that if the screen change ratio falls below the minimum (indicating lots of keystrokes with minimal screen update), an output anomaly is flagged.

16. The method of claim 12, wherein step (h) includes performing face recognition on the user images to authenticate identity, and if an unauthorized identity is detected in front of the computer (a face that is not the expected user), the method flags an anomaly or triggers a security action regardless of keystroke consistency, thereby protecting against scenarios where a different person attempts to carry out activities under another user's session.

17. The method of claim 12, further comprising a step of **temporarily halting** the delivery of keystroke inputs to the active application when an anomaly is flagged, and presenting a challenge to the user to verify human presence (such as a CAPTCHA test or a prompt requiring user reaction) before resuming, wherein failure to pass the challenge causes the session to be locked, thereby actively thwarting automated scripts in real time.

18. The method of claim 12, wherein the monitoring process runs in real-time during the user session and performs steps (e)–(j) continuously, so that an alert can be generated and transmitted substantially immediately upon detection of synthetic keystroke activity, and wherein in an alternate mode, the method buffers the data from steps (a)–(d) and performs steps (e)–(j) as a batch analysis after the session or after a fixed interval, providing a flexibility to operate either in live detection mode or forensic analysis mode.

19. **A non-transitory computer-readable medium** storing instructions that, when executed by one or more processors of a computing system, cause the system to perform the steps of the method of claim 12, thereby implementing a program for detecting synthetic or simulated keystroke activity through cross-referencing keystroke data with application context, screen content, and user presence information.

20. The computer-readable medium of claim 19, wherein the instructions corresponding to the anomaly aggregation and alert generation are further configured to allow configurable parameters to be set by an administrator, including thresholds for what constitutes anomalies in timing, context, and screen change, as well as enabling or disabling the facial verification requirement, thereby providing a customizable software solution that can be tailored to different operational environments and security policies.

21. The computer-readable medium of claim 19, wherein the instructions include a machine learning component that learns from feedback data consisting of historical alerts and user/administrator classifications of those alerts as true or false positives, and automatically adjusts the anomaly detection criteria over time to improve accuracy, wherein learned adjustments are applied to the analysis performed in steps (e)–(i) of claim 12 to reduce erroneous alerts and enhance detection of truly synthetic input patterns.

---

[1] [3] [4] [5] US9342687B2 - Detecting synthetic keystrokes - Google Patents

https://patents.google.com/patent/US9342687B2/en

[2] 14983978660405662 00-08763127

https://patentimages.storage.googleapis.com/11/b5/23/c3234fdd3bf94e/US8763127.pdf

[6] [7] [8] [9] CleverControl Employee Monitoring

https://clevercontrol.com/employee-monitoring/