Annexure3b- Complete filing

INVENTION DISCLOSURE FORM

1. **TITLE:** Self-Destructive ESP32-Based Keylogger (hardware) for Ethical Red Team Penetration Testing
2. **INTERNAL INVENTOR(S)/ STUDENT(S):**

|  |  |
| --- | --- |
| A. Full name | **Arindom Barman** |
| Mobile Number | 9394731853 |
| Email (personal) | [ritombarman81638@gmail.com](mailto:ritombarman81638@gmail.com) |
| Registration number | 12212440 |
| Address of Internal Inventors | School Of Computer Science and Engineering  Lovely Professional University, Punjab-144411, India |
| Signature (Mandatory) |  |
| B. Full name | **Viyanka Kamble** |
| Mobile Number | 9321749132 |
| Email (personal) | [viyankakamble0104@gmail.com](mailto:viyankakamble0104@gmail.com) |
| Registration number | 12216552 |
| Address of Internal Inventors | School Of Computer Science and Engineering  Lovely Professional University, Punjab-144411, India |
| Signature (Mandatory) |  |

## DESCRIPTION

### Illustration of Invention:

* 1. **Purpose**

The invention is a secret spy tool hidden as a normal USB stick. It does these things:

* Records key presses from a computer using USB
* Saves these logs safely in a small SD card
* Sends the stored data secretly over the TOR network using the computer's internet
* Spots any checks on it by watching power changes, USB oddities, and current draw changes
* Destroys both the data card and control chips via a small transformer, making sure no important info is left

### Technical Workings

When plugged in (best on Windows), the tool:

* Acts like a normal USB storage device to avoid being seen
* Starts logging keys in the background using USB
* Connects to the TOR network sometimes using the computer's internet and sends out logs and other data
* Always watches the power and USB flow for clues of checks
* If it sees meddling or strange things, it sends a high-voltage pulse through a small transformer, wrecking itself forever

### Internet of Things (IoT) Integration

The ESP32-WROOM-32 works mainly in secret but has tiny IoT features:

* Keeps hidden WiFi operations
* Sends system pings through safe TOR pathways
* Can be shut off or turned on remotely with pre-set safe codes

### Other Relevant Factors

* The build aims to be tiny, looking like regular USB sticks
* Keeping data safe matters most: logs are locked before saving or sending
* Two-step check stops accidental self-wreck unless clear signs of checks are seen

### Unique Attributes

* Secret physical hide.
* Smart self-wreck triggered by active checks
* Safe file sending over the TOR network using taken internet
* Locked storage and sending to stop data leaks even when it works normally

### Conclusion

This tool is a top-notch device for secret data gathering and keeping in risky spots. It works unseen and ensures data destruction when a threat is found, making it better than normal keyloggers and spy tools.

## DESCRIPTION OF THE DEVICE:

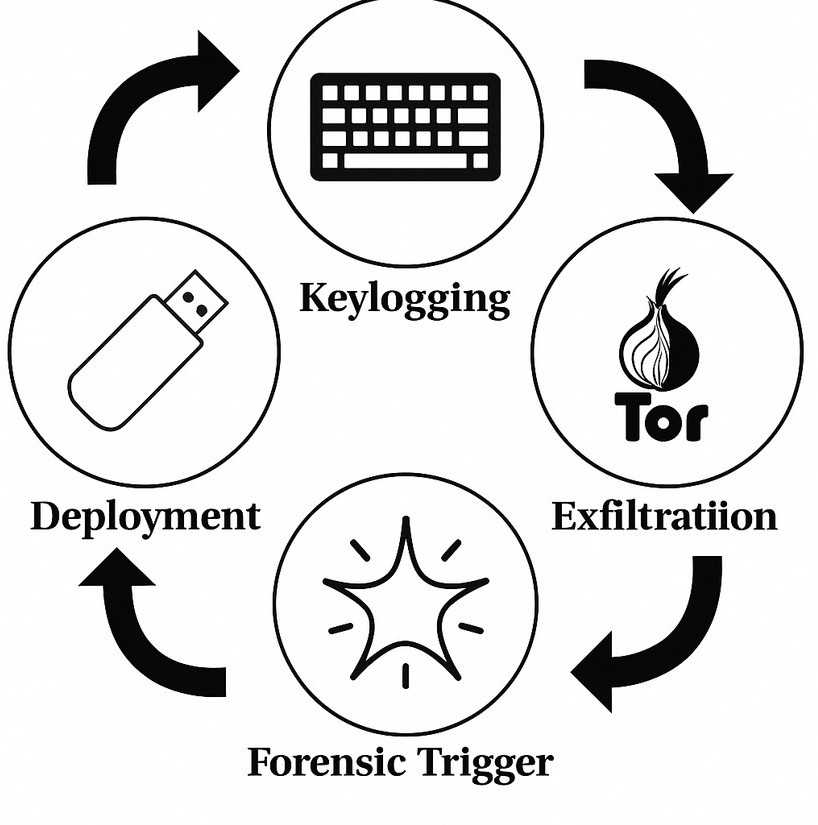
### Purpose

The device is a secret tool. It can take private data and type logs from a computer. It looks like a normal USB drive. Its main goals are:

* Collecting private things like passwords and files quietly.
* Sending data with the host’s internet using TOR to stay hidden.
* Destroying itself if found, so no one can recover data.
* Looking like a simple flash drive to avoid suspicion.

This device is good for tough digital places, spying on companies, or government missions where hiding is important.

### How It Works

****

* + 1. **Hardware**
       - ESP32-WROOM-32 Microcontroller: Runs the device, handles data, WiFi, and self-destruction
       - Micro SD Card (32 GB): Acts as storage for user and hides the captured data
       - USB Interface: Provides power (5V), data, and injects itself into the system
       - Self-Destruction Module: Uses a transformer and other parts to destroy itself when needed

### Firmware

* + - * Keylogging: The ESP32 captures typed keys on Windows using HID
      * Data Storage: Saves data encrypted on the SD card’s hidden part
      * Anonymized Transfer: Connects to the internet and sends data over TOR
      * Detection Mechanism:
        + Watches for voltage changes (USB sniffers)
        + Checks for odd USB activities
        + Monitors power use for strange behavior
      * Self-Destruction: If detected:
        + The transformer kicks in
        + High voltage destroys memory and electronics

### IoT Integration

The ESP32 can:

* Use WiFi in stealth to check-in with servers
* Get updates and commands securely
* Connect to a private TOR-based cloud for control
* Have a remote kill switch to destroy data fast

The device thus joins a network of spy tools, controlled remotely.

### Extra Points

* + 1. **Cross-Platform**
       - Starts with Windows but can be adapted to Linux and macOS

### Self-Learning (Future)

* + - * Could use AI to spot forensic checks from live USB data

### Power and Monitoring

* + - * Dual power modes: normal use draws low, mixed power, sensing danger makes it draw high power for destruction

### Stealth

* + - * Passive: Appears as a cheap slow flash drive.
      * Active: Uses fake scripts or decoy files to avoid suspicion.

### Unique Features

Feature Description

Self-Destruction on Detection Uses surges to kill hardware

Host Internet Exploitation Uses victim’s internet to connect over TOR HID Emulation Captures keystrokes without software

Real USB Drive Works as a normal flash drive

Multi-Channel Monitoring Monitors voltage, current, and USB for forensic analysis Remote Command IoT-enabled for managing actions

TOR-Based Anonymity Sends data through TOR for hiding Encrypted Storage Saves data encrypted in hidden parts

### Conclusion

This device blends stealth, hidden online actions, and irreversible self-destruction to make a top-notch spy tool. It’s useful for secret info gathering or business spying, and is a big step up from current keyloggers and data thieves.

### INTERNET OF THINGS (IoT) INTEGRATION:

* 1. **Talking to Host System**

The device uses the ESP32’s WiFi module, not as usual. It tricks the host PC (Windows OS) by USB HID imitation to:

* Make the host open secret TOR connections
* Insert scripts using Windows tools (like curl, wget, or PowerShell) to sneak data out
* Take over browser sessions or background services already linked to set up communication This way, no new network device shows up in IT tools, keeping it hidden

### Secret Data Transfer

The main way to sneak data out is TOR through Host Internet. Important points:

* No direct link between ESP32 and internet routers, hiding the device
* It uses Onion services to send logs to a hidden server
* All data is encrypted, keeping it unreadable even if caught by network sniffers Communications seem like regular encrypted traffic from the host system

### Standalone IoT Action

The device shows smart responses expected from an IoT system:

* Works Alone: Needs no human help once started.
* Event-Based Actions: Reacts to changes like voltage shifts, new requests, or power use changes.
* Safety Plans: If it can't sneak data out after many tries, it can destroy itself or try again later.

### Security and Backup

With its big task, the device uses many security layers:

* Triple AES Encryption: Data is locked at capture, stored, and during send-off.
* Backup Storage: If data sneaking fails, important data is copied to other parts of the SD card.
* Time-Based Self-Destruct: Can be set to destroy itself if offline for too long (e.g., 72 hours).

## EXTRA RELEVANT POINTS:

### Forensic Hiding

* No New Drivers: Uses usual HID and Mass Storage classes, avoiding suspicious drivers.
* No Lights or Sounds: Works quietly, with no status lights or sounds.
* Low Power Use: Uses power like a typical flash drive to avoid being noticed on checked USB ports.

### Power Use

* Saves Energy: ESP32 sleeps until the host is idle or unlocked.
* Controlled Power Surge: Only uses more power when self-destruct is triggered, avoiding early detection.

## UNIQUE ATTRIBUTES:

### Dual Roles

* Seen by People: Looks like a normal, safe 32GB flash drive.
* Smart Device: A hidden cyber weapon doing its job beneath the surface.

### Smart Self-Destruct

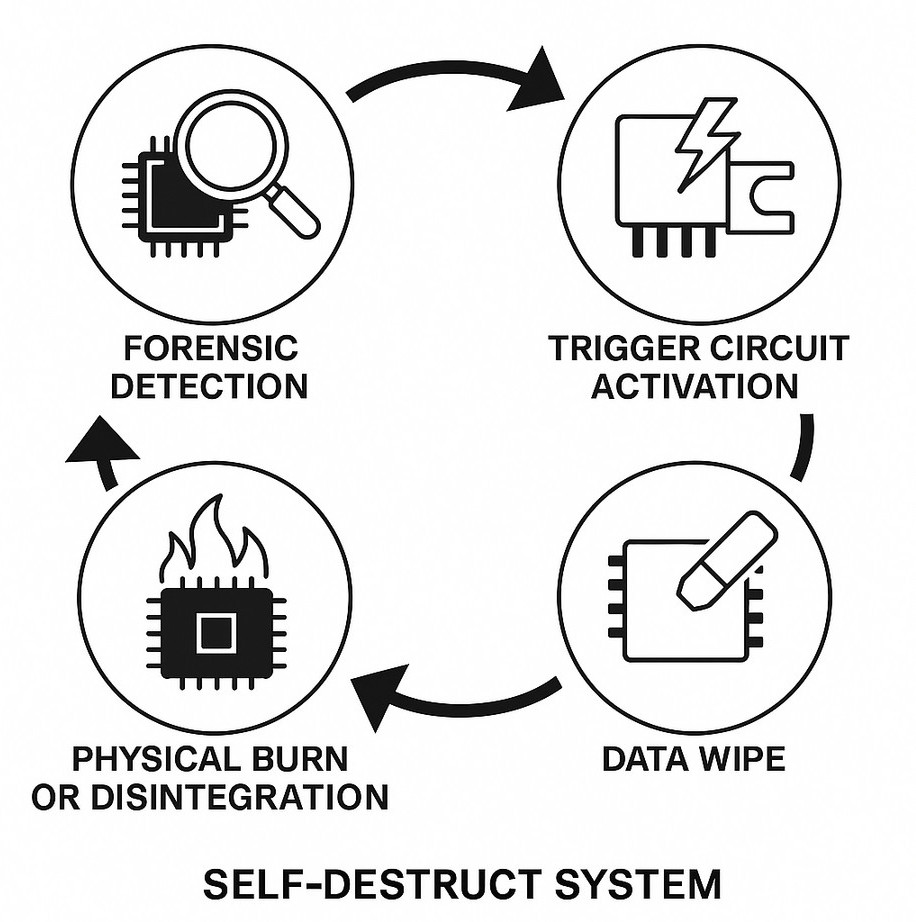
* Uses both setting triggers (USB signal changes) and remote commands to up chances of destroying itself before getting caught.

### Relies on Host Internet

* Only uses the host machine’s internet connection to avoid radio signals that could show it's there.

### Ultra-Hidden HID Insertions

* Quick Keyboard Imitation: Sends keystrokes quicker than a person can type, reducing exposure time.
  1. **Self – Destruct System Diagram**

****

## CONCLUSION:

This device marks a big step in secret cyber tools. By blending in like a flash drive, using smart IoT actions, secure hidden communication, and physical self-destruction, it brings a new type of cyber weapon that could change how spying is done in the digital age.

The device’s mix of stealth, smarts, and toughness makes it a strong tool in any cyber fight or intelligence-gathering task.

## PROBLEM ADDRESSED BY THE INVENTION:

1. Unauthorized Access: People can plug in bad USBs to secure computers.
2. Hidden Data Theft: Today’s keyloggers and USB sniffers are often found by anti-malware and network tools.
3. Slow Threat Detection: Standard tools may spot malware weeks later, giving attackers too much time.
4. Weak Protection: Typical keyloggers or attack USBs can be seized and studied if unplugged fast.
5. No Remote Wipe: If found, the device may risk losing the stolen files.
6. Risk of Analysis: Seized devices can be taken apart and studied, exposing secrets.
7. Loud Communication: Many bad devices use direct WiFi or GSM, which can be seen.
8. Small Storage: Regular USB keyloggers have low space, limiting data theft.
9. No Automatic Defense: Current devices don’t auto-detect probes like voltage or USB signal issues.
10. Hard to Scale: Manual setup and data grab is risky and slow.

Conclusion: This invention solves these key issues with real-time stealth, secret communication, and self-destruction in one small device.

## OBJECTIVE OF THE INVENTION:

1. Create a hidden device that looks like a flash drive.
2. Make it self-destruct when forensic tools are used.
3. Use host internet for safe, secret file transfer via TOR.
4. Allow remote updates, control, and self-destruct via dark web-based server.
5. Destroy memory parts to erase evidence.
6. Automate data grab, encryption, and theft with little user work.
7. Use cheap, common hardware (ESP32-WROOM-32 and SD cards).
8. Mimic regular USB flash drive behavior to stay hidden.
9. Add real-time decisions for self-destruction.
10. Enable wide deployment without raising detection risk.

## STATE OF THE ART / RESEARCH GAP / NOVELTY

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.**  **No.** | **Study** | **Abstract** | **Research Gap** | **Novelty** |
| 1 | Rubber Ducky (Hak5) | HID attacks using scripts | No forensic detection or self-destruction | Adds real-time forensic detection and hardware |

**Sr.**

**No.**

**Study Abstract Research Gap Novelty**

destruction

1. USBKill Software killswitch

for USB

Needs software installed on host

Hardware independent, works without host software

1. PoisonTap Hacks HTTP cookies Requires system

Works in real-time with

via USB

Record keystrokes

reboot

active session

Traditional Keyloggers

4

No self-destruction, easily detectable

Anonymized TOR upload, forensic evasion

NSA’s

1. COTTONMOUTH

Project

Covert USB espionage tool

Extremely high cost and complexity

Low-cost open-source equivalent

1. BadUSB Exploits Reprogramming USB Needs customized

ESP32 emulation without

firmware

1. USB Condoms Block unauthorized

USB devices

USB firmware

No self-detection in payloads

reprogramming firmware

Device fakes authorized profiles dynamically

Malware-in-USB research

8

Malware planted in USBs

Easily flagged by AV Encrypted payloads and

1. TOR Botnets Use TOR for C2 Need persistent

stealth behavior

malware installed

No software left behind

1. USB Firewall Devices Monitor USB ports Detect on connection Dynamic connection profile

switching

### Conclusion:

This little device fills a major gap in the cybersecurity world. It’s a clever mix of stealthy communication, real-time threat detection, and physical self-destruction, all packed into something that looks just like an ordinary USB flash drive.

## DETAILED DESCRIPTION:

### Device Bits:

ESP32 unit, 32GB MicroSD card, control chip, voltage booster, small parts.

### Look:

Housed in normal flash drive shell; heat sinks near booster.

### Power:

Works on 5V from USB.

### Storage:

SD card for normal files plus secret logs.

### Network Use:

Uses host's session to go on TOR; no direct WiFi.

### Forensic System:

Checks voltage changes, USB patterns, current jumps.

### Self-Destruct:

Voltage booster makes bursts of >30V to damage SD card.

### Data:

AES-encrypted before saving or sending.

### Control Chat:

Asks hidden TOR server for commands often.

### Safety:

Failsafe destruction if offline long or can't reach server.

### Conclusion:

The gizmo is a full rogue USB with built-in computer defense and self-destruct.

## SYSTEM COMPONENTS

1. ESP32 Unit:

Main part, can WiFi chat, act as USB (using TinyUSB), talk to SD card.

1. MicroSD Storage:

At least 32GB storage; works like normal flash drive while hiding logs and scripts.

1. USB Plug (Type-A Male):

Connects to host machine, gives 5V power, handles USB data chatting.

1. Voltage and Current Checking:

Custom parts check USB VBUS and D+ / D- line talk for changes linked to probing or tools.

1. Voltage Booster and Destroy:

Circuit to boost 5V to around 100V (or enough to fry SD card) when triggered.

1. Fake Casing:

3D-printed or molded to look like normal flash drive to avoid notice.

1. Passive Cooling:

Metal inner casing or heat-moving stuff to cool during normal use and heat fast during destruct.

1. Firmware:

Software on ESP32 to manage logging, encrypting, forensic checking, TOR chatting, and self-destruct.

1. Data Encryption:

Light (e.g., AES-128) encryption of logs before sending to keep secret even if data is taken.

1. TOR Chat Module:

Built-in ways to send data secretly through host's net without detection.

### Conclusion:

Each part is key to make the gizmo work, stay hidden, and be solid under checks.



### IoT SENSORS AND CHAT

1. USB Signal Three-Way Change:

ESP32 inner ADCs check odd changes in USB D+/D- signaling to spot hostile probing.

1. Voltage Watching:

Constant checks of 5V VBUS line for strange drops or spikes linked to forensic attacks.

1. Current Use Look:

Checking parts measure current use now, flagging odd values as threats.

1. WiFi Scan:

ESP32 can quietly scan nearby networks to find open spots or check the host's net link.

1. Host Chat Takeover:

Using USB network profiles to use host TCP/IP stack for net use without raising alarms.

1. TOR Use:

Secured chat with hidden onion servers to send data with low trace.

1. Passive Event Watch:

Firmware listens to USB “keep-alive” packets and talk times to see if checks are happening.

1. Host Activity Watch:

Logging host idle times (no keyboard use) to pick safe times to send bursts.

1. Data Send (Optional):

After key data is taken, optional HID payloads can be sent (like opening a command line and self- wipe).

1. Triggered Self-Destruct:

All sensors feed into a choice tree. Certain limits automatically trigger self-destruct rules.

### Conclusion:

With many sensors, the gizmo turns into a smart, reactive agent able to guard itself in tough spots.

## MAIN SYSTEM CONTROL

1. On-Device Brain:

ESP32 runs rules to know when to get data, send, or blow up.

1. Take in Data System:

Keys or files saved first in locked bits before sending.

1. TOR Tunnels:

Makes hidden paths for data to go out without being seen.

1. Blow-Up Handler:

Watches power and signals to see if it needs to blow up.

1. Code Update Handler:

Can check hidden TOR sites for updates.

1. LEDs (Low-Key):

Tiny LEDs inside for checks during build.

1. Data Check:

Checks data to make sure it's right before blow-up (to avoid loss).

1. Power Control:

Saves power by turning off WiFi when idle.

1. Lock-up System:

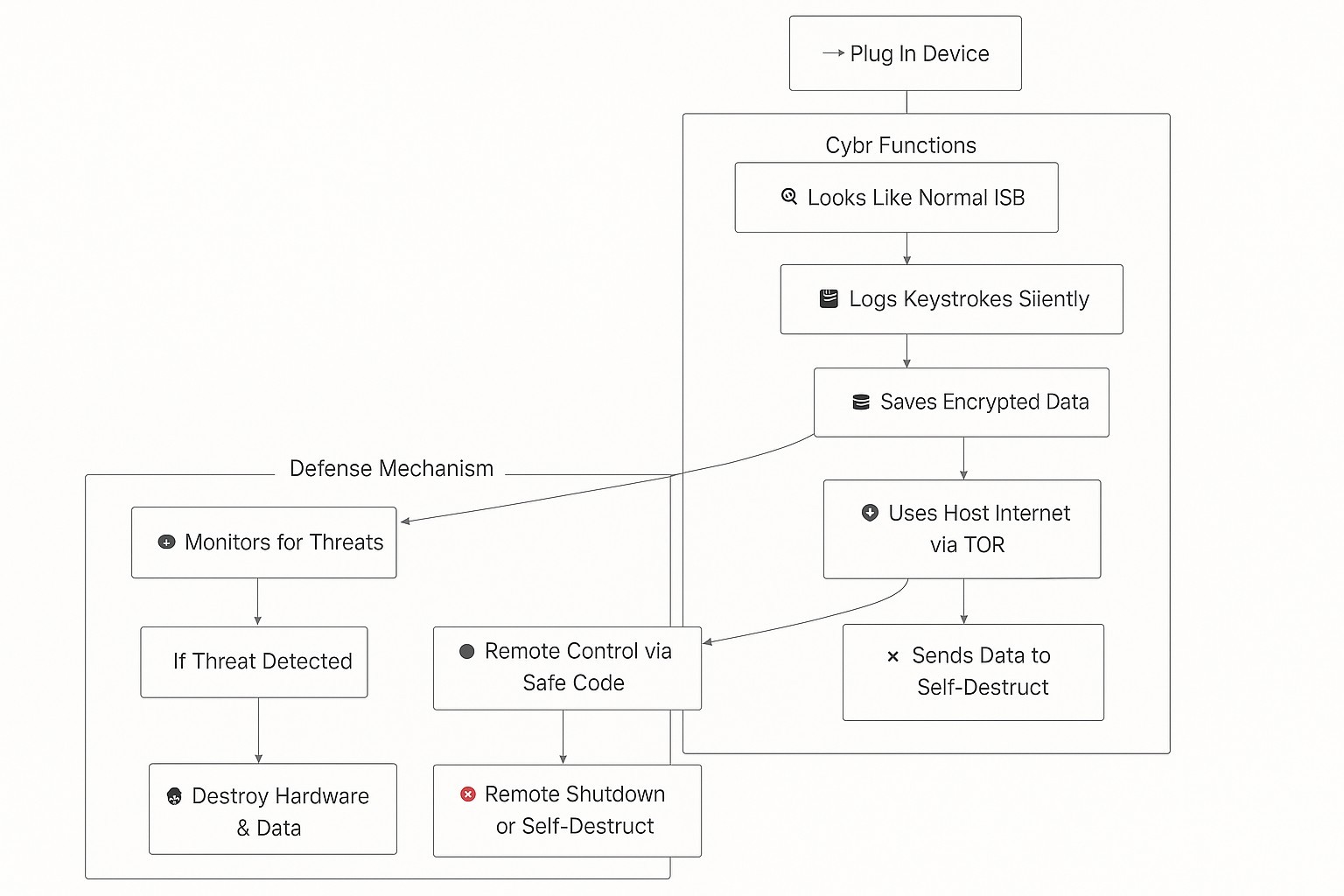
Makes keys each time and keeps them only in quick-use memory.

1. Wipe-on-Catch:

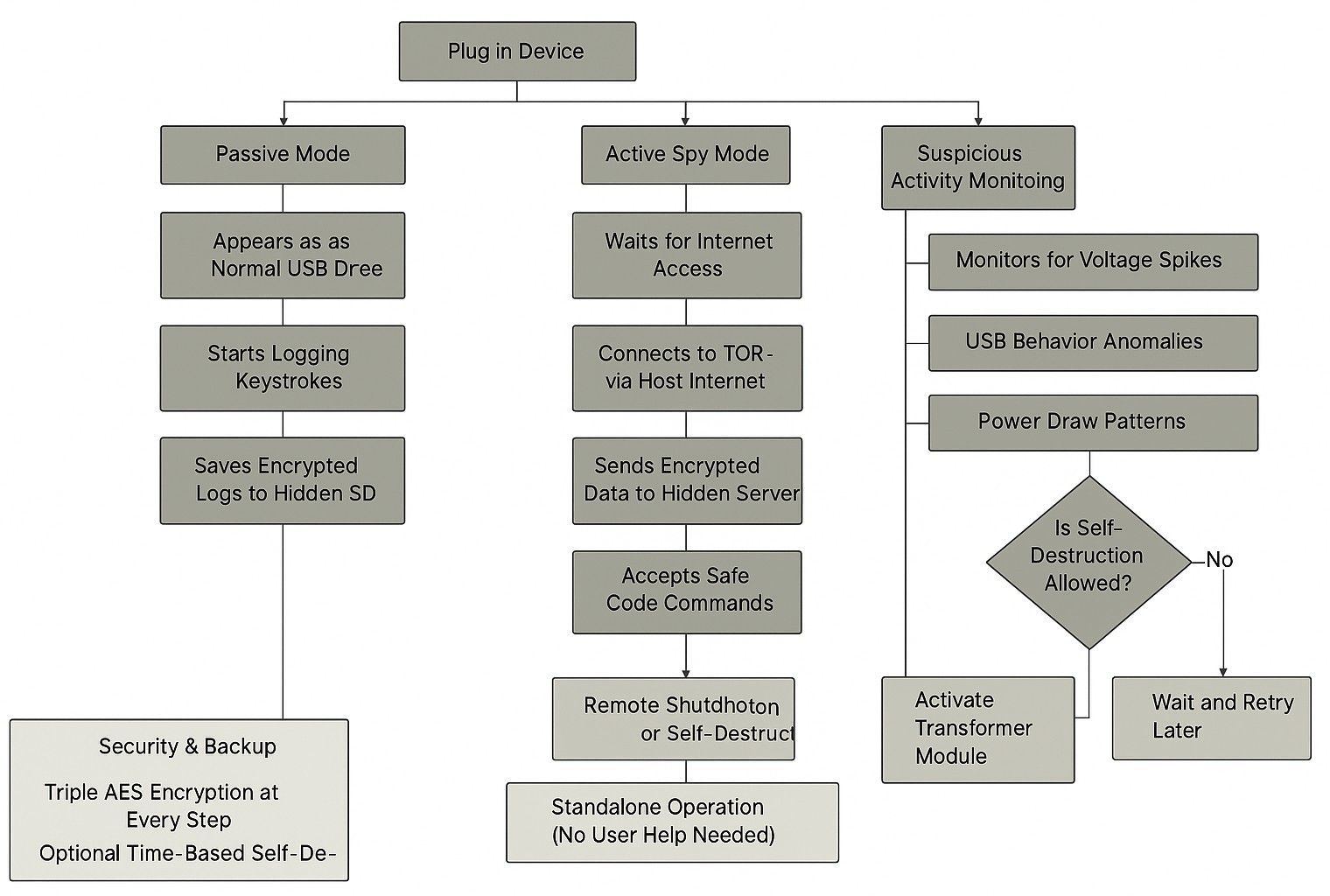
If plugged into bad ports (e.g., checker PCs), erases all memory fast.

End:

Makes the tool work alone as a spy gear.



**Process Workflow:**

****

## HIDDEN TECH

1. Quiet Key Record via HID:

Looks like a keyboard, gets keys or sends commands in secret.

1. SD Card Store:

Logs and data kept on a hidden part of the card - users can't see.

1. Sees Odd USB Action:

Watches USB lines for things like power change or too many checks.

1. TOR Link:

Uses PC internet to link TOR for sending data.

1. Blow-Up from Power Spike:

If bad action is seen, turns on to break the memory chip.

1. Data Lock Before Sending:

Data locked at the start to stop others from seeing it.

1. Signal Check Match:

Looks for known signs of checker tools.

1. PC System Check:

Works out if it's in a Windows, Linux, or Mac PC and changes behavior (Windows is main goal).

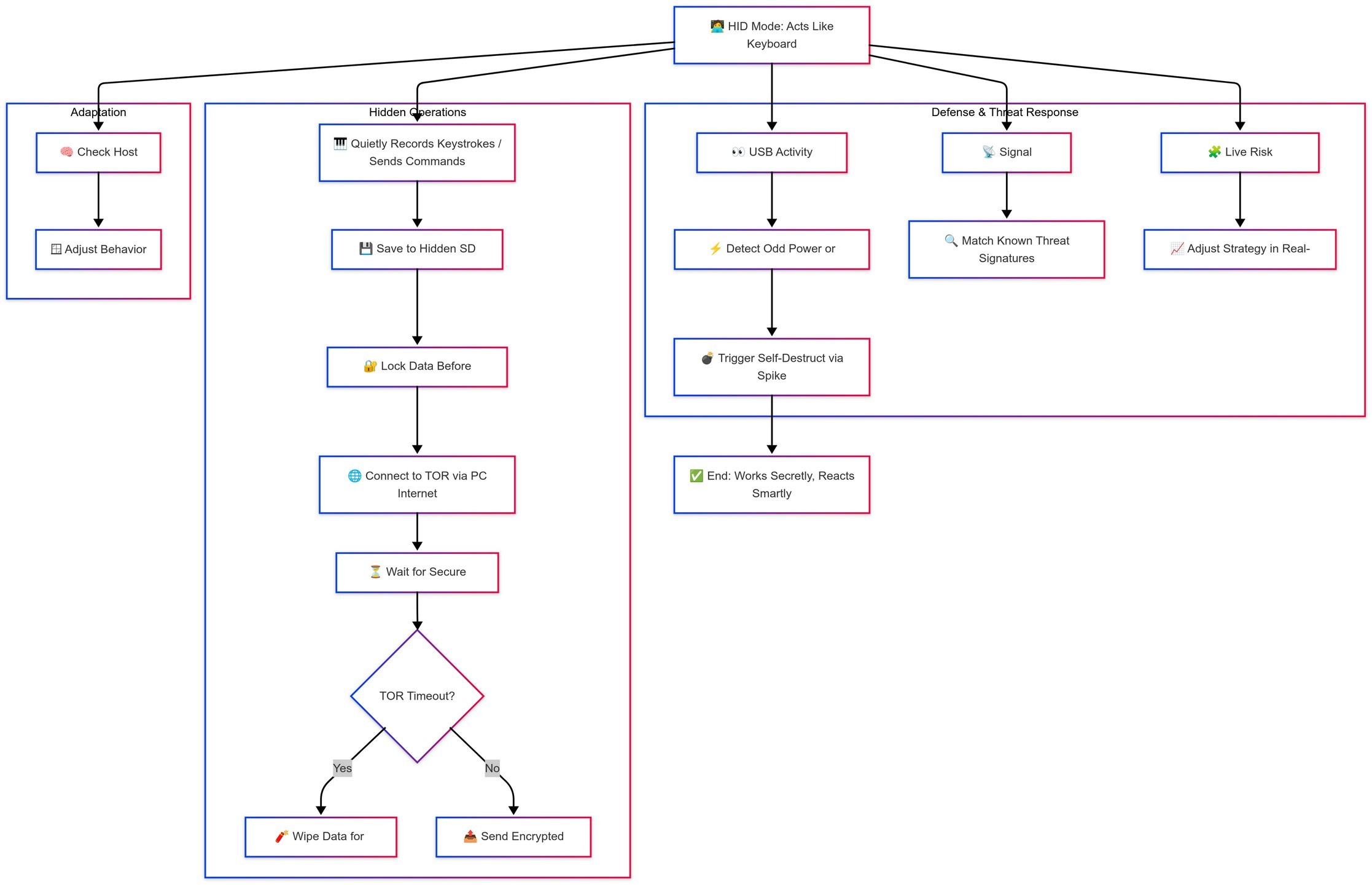
1. Live Threat Check:

Keeps checking risk and adjusts actions.

1. Timeout Safety:

If no TOR link in time, can choose to wipe data. End:

Tech setup aims for secret, smart work making the spy tool hard to find.



## LIVE DATA CONTROL

1. Instant Key Take:

Each key press logged with time, keeping event right.

1. Wait to Write:

Waits to write to SD card to avoid wear.

1. Log Extra Data:

Logs keys and also states, window titles, and user stuff.

1. Data Shrink:

Data made smaller before locking to save space.

1. Error Fix:

Checks and can fix data before sending or storing.

1. Quiet Sync:

Data sent out when internet found, in secret.

1. Masked Data:

All link-related data is hidden to cut risks.

1. Forensic Proof:

Writes spread out on card to stop recovery.

1. Fast Data Dump:

If attack seen, all data sent out, then starts blow-up.

1. Clean Schedule:

Old data wiped with many passes making sure nothing remains. End:

Live data work keeps logs and cuts risk of getting caught.

## SPECIAL PARTS

1. Blow-Up Action:

Has small tool to break SD card and memory if tampered with.

1. Dual USB Trick (Store + HID):

Looks like both a drive and a keyboard, logs keys without notice.

1. Power Watch:

Checks power and data lines nonstop, triggers defense if odd things are seen.

1. Hidden Data Send:

Uses TOR to send data, keeping it untraceable.

1. Quiet Work:

Does not show up, only sends data in safe times, cutting detect risk.

1. Wipe on Alone:

If cut from network fast, erases all memory and gets ready to blow up.

1. Smart Threat Check:

Not all odd things mean blow-up. Judges if need is there.

1. Secret Look:

Looks like an old drive, can't be picked out from many others.

1. Locked Inside Talk:

Inside talk and data locked to stop attacks.

1. Live PC Check:

While on, keeps checking PC (OS version, networks, active apps) to spot checks or safe spaces.

## GAINS AND PLUSES

1. Hidden Path:

Thanks to TOR, data send has no trace back to owner.

1. High Stay Rate:

Only top checks may find it - still, blow-up means no data left.

1. Many Use Cases:

Can be used for tests, secret missions, or alert actions for safe data grab and send.

1. No Special Needs:

As it uses common USB rules (HID and Store), PC needs no extra drivers.

1. Fast Data Kill:

Kill is real, not just a fake wipe, advanced tools can't get data back.

1. Power Safe:

Even if checks watch power use, runs like a common flash drive till triggered.

1. Works on Many PCs:

Best on Windows, but can run on Mac and Linux too.

1. Cheap Build:

Uses ESP32 WROOM-32, easy PCB, small SD, and step-up tool, affordable unlike military gear.

1. Easy Use:

Once in, it runs alone, no user action needed.

1. Small Trace:

Leaves nothing on the PC, no remains even with fast checks.

## EXPANSION

1. Add Bluetooth LE (BLE) Command Channel: Let nearby operator trigger self-destruction or silent shutdown remotely.
2. Integration with Cellular Modules: Device sends data over 4G/5G, skipping host’s internet.
3. Advanced Forensic Detection Algorithms: Use machine learning in ESP32 to spot subtle forensic attacks or sandboxes.
4. Camouflaged Form Factors: Make the device look like coins, pens, or key fobs to avoid suspicion.
5. Multi-Step Destruction: Destroy flash storage and break key microcontroller paths to stop hardware reverse-engineering.
6. Self-Healing Functions: After a small threat, wipe only recent data, not all.
7. On-Device AI Analysis: Check keystrokes locally to spot patterns (e.g., passwords vs. random typing) and decide which data to send first.
8. Fake Data Output: During forensic check, show fake harmless files to mislead.
9. Battery Backup: Small battery or capacitor ensures self-destruction even if USB power cuts off.
10. Cluster Deployment: Distribute many devices across machines to catch data and destroy in sync if one gets compromised.

## WORKING PROTOTYPE / DESIGN / COMPOSITION

1. Microcontroller: ESP32 WROOM-32 with Dual-core, 240 MHz, 4 MB Flash, WiFi, and BLE capability.
2. Storage: MicroSD card module, at least 32GB, write cycle optimized.
3. USB Interface: CP2102N or native USB OTG from ESP32 for emulating HID and Mass Storage.
4. Power Circuit: 5V USB input, with boost circuit generating 15V-20V for physical destruction.
5. Sensors:
   * Current sensor (INA219 or simple shunt resistor)
   * Voltage monitor (via internal ADC)
   * Passive USB data sniffer (USB differential line tap)
6. Stealth Casing: Old USB drive shell, metallic shielding inside to block RF emissions.
7. Firmware:
   * FreeRTOS multi-threaded application
   * Custom USB stack for HID+Mass Storage mode
   * Lightweight AES encryption
   * TOR client for ESP32
8. Destruction Circuit: Triggered by GPIO pins, with voltage detect thresholds, activating transformer via MOSFET switch.
9. Testing Setup: Use forensic USB tools (e.g., USBlyzer, Wireshark USBPcap) to simulate attacks and adjust detection.
10. Prototype Output: Small, thumb-sized USB stick for stealth, test in real settings like offices

## EXISTING DATA

* + WiFi and Networking: ESP32 has TCP/IP stack, handles encrypted WiFi comms, proxies, TOR hidden services.
  + SD Card Protocols: MicroSD cards use SPI with robust libraries for read/write.
  + Step-Up Transformer Working: Small DC-DC converters (boost) raise 5V USB to 15V–20V, burning out SD card circuits.
  + USB Protocol Basics: USB Mass Storage uses SCSI over Bulk transfers, HID uses Interrupt transfers

— emulatable by ESP32 libraries.

* + Forensic USB Analysis: Forensic attacks involve descriptor requests, voltage surges, and current draws, detectable by monitors.

### Conclusion:

Building this device needs skills in microcontroller firmware, electrical design, cybersecurity, and forensic countermeasures to make a true "ghost device" that stays hidden and self-destructs if caught.

## USE AND DISCLOSURE (IMPORTANT):

Please answer the following questions:

|  |  |  |
| --- | --- | --- |
| A. Have you described or shown your invention/ design to anyone or in any conference? | YES( ) | NO ( X ) |
| B. Have you made any attempts to commercialize your invention (for example,  have you approached any companies about purchasing or manufacturing your | YES( ) | NO ( X ) |

|  |  |  |
| --- | --- | --- |
| invention)? |  |  |
| C. Has your invention been described in any printed publication, or any other form of media, such as the Internet? | YES( ) | NO ( X ) |
| D. Do you have any collaboration with any other institute or organization on the same? Provide name and other details. | YES( ) | NO ( X ) |
| E. Name of Regulatory body or any other approvals if required. | YES( ) | NO ( X ) |

1. Provide links and dates for such actions if the information has been made public (Google, research papers, YouTube videos, etc.) before sharing with us. **NA**
2. Provide the terms and conditions of the MOU also if the work is done in collaboration within or outside university (Any Industry, other Universities, or any other entity). **NA**
3. Potential Chances of Commercialization. **Yes**
4. List of companies which can be contacted for commercialization along with the website link.

Here are two companies that specialize in electric vehicle charging solutions and could be potential partners for the commercialization of the Self-Sufficient Charging System for Electric Public Transportation:

(List the companies which maybe interested)

1. Any basic patent which has been used and we need to pay royalty to them.
2. FILING OPTIONS: Please indicate the level of your work which can be considered for provisional/ complete/ PCT filings - (Provisional)

### Keywords

ESP32-WROOM-32

USB HID Emulation Mass Storage Emulation Keylogger

Stealth Device

Self-Destruction Mechanism Voltage Fluctuation Detection

Current Draw Anomalies

TOR Network Communication Anonymous Data Exfiltration Step-Up Transformer Encrypted Data Storage

Real-Time Host Profiling Forensic Detection Passive Network Sniffing Auto-Wipe Mechanism Low-Power Operation Threat Level Assessment Stealth Flash Drive Windows Target System

## CONSENT FORM FOR ORIGINALITY OF IDEA

**Name of Inventor:** Viyanka Kamble , Arindom Barman

**RegID:** [12216552] [12212440]

**School/Department:** School of Computer Science and Engineering

**Program:** B.Tech (CSE) with Minor in Cybersecurity

**Domain Area:** Cybersecurity, Ethical Hacking, Embedded Hardware Systems

**Date:** 10/04/20225

# Title of the Idea :

**Self-Destructive ESP32-Based Keylogger (Hardware) for Ethical Red Team Penetration Testing**

# Declaration of Originality

We, hereby solemnly affirm and declare that the above-mentioned idea and its conceptualization, design, and implementation are original, novel, and created solely by us. This idea has been conceived with academic, ethical, and innovative intentions to contribute to the domain of penetration testing under red team operations.

The core concept involves the design and prototyping of a self-destructive hardware keylogger using an ESP32 microcontroller. The proposed invention includes functionalities such as encrypted data storage on an SD card, wireless data transmission, and a secure self-destruct mechanism via a controlled high- voltage surge to ensure data irrecoverability upon unauthorized access.

# Affirmations:

### Original Thought Process

We confirm that the idea was developed through personal learning, research, experimentation, and application of cybersecurity and embedded system knowledge. It is not directly copied or derived from any published work or existing product.

### Patent Novelty Check

A comprehensive patent novelty search using Google Patents and other databases has been conducted by us to ensure that the proposed design and application of a self-destructive ESP32-based keylogger is not available or filed under any similar intellectual property.

### No Infringement

We declare that this project does not infringe upon any known intellectual property rights or third-party copyrights, patents, or trademarks. If any related or prior art has been referred to, it has been appropriately acknowledged in documentation.

### Purpose and Ethics

The invention is designed strictly for **ethical use** in red team penetration testing environments

for cybersecurity audits, education, and research purposes. It is not intended for unethical, malicious, or unauthorized surveillance or data capture.

### Responsibility and Acknowledgment

We take full responsibility for the authenticity and technical accuracy of this idea. Any future developments or collaborative enhancements to the project will be duly acknowledged and documented.

# Inventor Declaration:

We understand that if this idea is found to be plagiarized or not original at any stage of review, evaluation, or patent process, necessary action may be taken which may include disqualification of the patent submission and academic penalties.

**Inventor’s Name:** Viyanka Kamble, Arindom Barman

**Reg no:** [12216552] [12212440]

**Date:** 14th April, 2025