“Design and Implementation of a Secure Battery Management System (BMS) with Safety Features for Lithium-based Batteries”

Sumith Subraya Nayak sumithnayak07@gmail.com 7019685443

Problem statement:

The original equipment manufacturer (OEM) provided the firmware and pager application to the spy agency. In turn, the agency was able to modify the firmware of the Battery Management System (BMS), disabling the built-in safety mechanisms for lithium-based batteries. This compromise allowed for thermal runaway, ultimately leading to an explosion and resulting in casualties.

WORLD

The Tribune

CHANDIGARH | WEDNESDAY | 18 SEPTEMBER 2024

ead, over 2,700 hurt in pager onations across Lebanon

EPTEMBER 17
ed and around
ded, with 200
on, as pagers
ah members
nultaneously
Health Minis-
aid on Tues-
f explosions
our after the
, which took
n. It was not
r how the
ated.
ded several
ers and
assador to
Amani, also
jury when
an's semi-
s agency
eign Min-
hi, in a
Lebanese
gly con-
st attack"

Iran's envoy hurt, Hezbollah blames Israel

Ambulances arrive at a Beirut medical centre as thousands were wounded after pagers exploded in Lebanon on Tuesday. REUTERS

and thanked Lebanon for providing immediate medical treatment to Amani.

A Hezbollah official, speaking on condition of anonymity, said the detonation of the pagers was the "biggest security breach" the group had been subjected to in nearly a year of conflict with Israel.

Israel and the Iran-backed Hezbollah have been engaged in cross-border warfare in parallel with the Gaza war which erupted last October. The Israeli military declined to comment as Reuters enquired about the detonations.

Lebanon's information Minister Ziad Makary said the government condemned the detonation of the pagers as an "Israeli aggression". Hezbollah also blamed Israel for the pager blasts and said it would receive "its fair punishment". Besides, Hezbollah chief Hassan Nasrallah was not harmed in the spree of pager blasts across Lebanon, a senior Hezbollah source told Reuters on Tuesday.

The Israeli military declined to comment on Reuters enquiries about the detonations. — Reuters

.**Study of Lithium-based Battery Failures:**

- Analyze how lithium batteries fail, focusing on thermal runaway, and the role of BMS in preventing such incidents.

2.**Firmware Security for BMS:**

- Investigate how unauthorized firmware updates could bypass safety features.
- Propose a secure firmware update mechanism using encryption and authentication methods.

3.**Implementation of Safety Protocols:**

- Develop a prototype BMS that includes safety protocols (overvoltage, overcurrent protection).
- Simulate conditions that could lead to thermal runaway and demonstrate how the BMS mitigates such risks.

4.**Testing and Validation:**

- Test the system under different load and temperature conditions.
- Validate the security measures against potential attacks on the firmware.

Tools & Components:

- Microcontroller (e.g., Arduino, ESP32)
- Lithium-ion battery cells
- Temperature and voltage sensors
- Battery protection circuits
- Firmware development platform (e.g., STM32CubeIDE)

To successfully carry out our project, which focuses on lithium-based battery failures, BMS security, and firmware integrity, you'll need both hardware and software tools for design, simulation, and testing. Here's a breakdown of the tools and software you'll need at each stage:

1. Study of Lithium-based Battery Failures:

- **Battery Simulation Software:**
  - **COMSOL Multiphysics**: For simulating thermal and electrochemical behaviors of batteries. It can model heat generation and dissipation in lithium batteries under different conditions.
  - **MATLAB/Simulink**: Useful for modeling and simulating battery dynamics, failure modes, and the impact of various operating conditions.
  - **LTspice**: A free, powerful tool for simulating electrical circuits, including battery systems.

2. Firmware Security for BMS:

- **Embedded Systems Development IDE:**
  - **STM32CubeIDE (for STM32 microcontrollers)**: Popular for writing, simulating, and debugging firmware for BMS. Offers a complete development ecosystem for secure firmware development.
  - **Keil uVision**: Widely used for ARM-based microcontroller firmware development, ideal for secure firmware updates and embedded security protocols.
- **Firmware Security Tools:**
  - **OpenSSL**: For implementing encryption and authentication methods to ensure secure firmware updates.
  - **Atmel Studio**: If you are using Atmel microcontrollers (e.g., ATmega), this tool can help in designing and testing secure firmware.

3. Implementation of Safety Protocols:

- **Microcontroller Programming Software:**
  - **Arduino IDE**: If you are prototyping with Arduino boards, the Arduino IDE will allow you to write and upload firmware that includes safety protocols (overvoltage, overcurrent, thermal protection).
  - **PlatformIO**: An alternative development platform for various microcontroller boards (ESP32, STM32, etc.) with integrated security and firmware management features.
- **Circuit Design and Simulation Software:**
  - **Proteus Design Suite**: Can be used to design and simulate electronic circuits, including the integration of a BMS and safety features.
  - **Altium Designer**: For more advanced PCB design and simulation, useful if you plan to develop a custom BMS.

. Testing and Validation:

- **Battery Testing Tools:**
  - **Battery Test Equipment**: Such as battery testers (Arbin, Chroma) that can simulate different load and temperature conditions on the physical battery and BMS.
- **Security Testing Tools:**
  - **Wireshark**: For monitoring data traffic and verifying secure communication protocols between the BMS and external devices.
  - **Penetration Testing Software**: Tools like **Metasploit** can be used to test for vulnerabilities in the firmware security.
- **Thermal Monitoring Software:**
  - **FLIR Tools**: If you have access to a thermal camera, you can use FLIR software to monitor and validate the temperature profiles of the battery during testing.
- **MATLAB/Simulink**: For analyzing results from load and temperature testing, and validating how well the BMS responds to thermal runaway conditions.

Hardware Requirements:

- **Development Boards**:
  - Arduino or STM32 boards for prototyping.
  - Sensors (temperature, voltage, current) to interface with your microcontroller.
- **Lithium-ion Batteries**: For real-world testing.
- **Battery Management System (BMS)**: Off-the-shelf or custom BMS boards.

Steps to Implement:

1. **Model and simulate the lithium battery and failure scenarios using COMSOL/Simulink.**
2. **Develop and secure the BMS firmware in STM32CubeIDE or Arduino IDE.**
3. **Use Proteus or Altium for designing circuits that will integrate the BMS and sensors.**
4. **Simulate security scenarios and firmware attacks using OpenSSL and testing tools like Metasploit.**
5. **Test the physical battery and BMS under load and varying temperature conditions using battery testing equipment.**

This combination of tools will allow you to cover all stages of the project from design and simulation to security and testing.