CHAPTER I
INTRODUCTION

1.1 Background

Early Warning Systems (EWS) play a vital role in reducing risk and minimizing damage during disasters. With the growth of the Internet of Things (IoT), EWS can now support real-time monitoring and automated alerting, enabling faster and more effective responses. Most IoT-based EWS solutions have focused on improving data transmission and latency, but aspect such as secure key exchange, memory usage, and CPU load are still not fully explored.

Several communication protocols, such as Message Queuing Telemetry Transport (MQTT) and Google Remote Procedure Calls (gRPC), have been used to enhance energy efficiency and communication reliability in EWS deployments [1–3]. Some systems, like QuakeSense [4], use LoRa for long-range seismic monitoring. Traditional security approaches, including Transport Layer Security (TLS) and Software Defined Network (SDN), have been widely applied to IoT [5–7]. While effective, TLS relies on certificates and centralized Public Key Infrastructure (PKI), resulting in additional communication overhead, more complex key management, and higher memory and CPU usage challenges for resource-constrained, latency-sensitive EWS environments.

Hierarchical Identity-Based Encryption (HIBE), especially the Boneh-Boyen (BB-HIBE) scheme, provides an alternative to certificate-based approaches. Without digital certificates, and with the benefit of hierarchical key delegation, BB-HIBE makes key management simpler and helps cut down on communication and storage requirements. More recent work has also looked into threshold and revocable HIBE, which aim to make key management and access control in large IoT systems even more flexible [8,9]. These developments highlight the increasing move toward certificate-less security for key distribution.

Even with these advancements, most studies so far have focused on general key management or access control, rather than evaluating HIBE specifically for the key exchange process in real EWS scenarios. In this research, the author focuses on how BB-HIBE can be utilized for secure key exchange in IoT-based EWS, considering communication overhead, key storage, and performance in relation to international standards. Most previous studies have overlooked these practical aspects. Although

BB-HIBE has clear theoretical advantages, its actual performance in EWS and IoT contexts remains underexplored. This research addresses that gap by:

- Comparing the scalability of BB-HIBE and standard key exchange methods (TLS/ECDH) in terms of communication overhead and key storage,
- Evaluating BB-HIBE's computational performance at 128-bit and 256-bit security levels, including execution time, CPU usage, and memory usage,
- Benchmarking all results against international IoT standards for latency ($<$100 ms), CPU use, and memory requirements.

The simulation results presented in this research demonstrate that BB-HIBE is practical for key exchange in EWS applications, and its hierarchical, certificate-less structure makes it well-suited for secure, scalable, and resource-constrained IoT environments.

1.2 Problem Identification

Current Early Warning Systems (EWS) generally use traditional security mechanisms such as TLS and certificate-based approaches during the initial key exchange. However, these methods introduce notable overheads, increasing both computational load and communication latency, which are critical constraints in resource-limited and latency-sensitive IoT deployments. To overcome these challenges, a lightweight cryptographic solution is required to protect key exchange without significantly affecting system responsiveness and resource efficiency.

1.3 Research Objective

Based on the problem outlined above, the research objectives are formulated as follows:

1. To propose and evaluate the Boneh-Boyen Hierarchical Identity-Based Encryption (BB-HIBE) scheme specifically for the key exchange mechanism within IoT-based Early Warning Systems, aligned with IEEE Std 1363.3-2013 [10].
2. To quantitatively assess the BB-HIBE scheme's performance, particularly focusing on computational latency (execution time), CPU usage, and memory usage at different security levels (128-bit and 256-bit key sizes).

3. To perform a comparative analysis of BB-HIBE versus the traditional TLS/ECDH mechanism in terms of communication overhead and total key storage requirements, ensuring compliance with international IoT security and performance standards.

1.4 Scope of Work

To maintain clarity and focus, this research is limited to the following scope:

1. The implementation and evaluation of the BB-HIBE scheme will strictly adhere to the IEEE Std 1363.3-2013 guidelines [10].
2. The evaluation will focus exclusively on the key exchange phase during the initial handshake process, not covering the entire communication session.
3. Simulation and performance analysis will be conducted in a Microsoft Azure Virtual Machine environment, using Python programming language along with the Charm-Crypto library.
4. The simulation scenarios include measuring key exchange efficiency in terms of communication overhead, key storage requirements, execution time, CPU utilization, and memory consumption for 128-bit and 256-bit security levels.

1.5 Hypothesis

The hypothesis proposed in this research is that the Boneh-Boyen HIBE scheme can effectively serve as an alternative key exchange mechanism for IoT-based Early Warning Systems. The BB-HIBE scheme, due to its certificateless architecture and hierarchical key delegation capability, is expected to substantially reduce communication overhead, storage demands, and computational complexity compared to conventional methods (TLS/ECDH). Moreover, the BB-HIBE implementation with a 128-bit security level is anticipated to satisfy international performance standards (IEEE and NIST) related to latency, CPU usage, and memory consumption, making it practical and secure for real-time deployments in EWS environments.

1.6 Methodology

This research employs a structured and quantitative approach involving the following stages:

1. System Design: Develop a hierarchical key exchange model using BB-HIBE specifically tailored for the IoT-based Early Warning System scenario.

2. Implementation: Build the proposed cryptographic scheme in a simulated Microsoft Azure environment, using Python integrated with the Charm-Crypto cryptographic library.
3. Simulation and Measurement: Conduct comprehensive simulations to obtain data on execution times, CPU and memory usage, as well as communication overhead and key storage needs, comparing BB-HIBE against TLS/ECDH.
4. Result Analysis: Analyze the collected simulation data, benchmark the results against existing international standards, and draw conclusions regarding BB-HIBE's practical applicability.

1.7 Research Methodology

To achieve the stated research objectives, the following methodological framework is adopted:

1. Literature Study: Review existing studies related to key exchange protocols, IoT security requirements, identity-based cryptography (IBE/HIBE), and relevant international standards.
2. System Modeling: Develop a practical and realistic model for hierarchical key distribution and management using the BB-HIBE scheme.
3. Implementation and Simulation: Conduct practical simulations under multiple scenarios (communication overhead, key storage, execution time, CPU, and memory usage) within a controlled environment (Microsoft Azure VM).
4. Evaluation: Evaluate the BB-HIBE scheme's performance based on predefined metrics and criteria, comparing results against the standard TLS/ECDH scheme and international performance guidelines.
5. Conclusion and Recommendations: Summarize research findings, highlight the advantages and limitations of BB-HIBE, and provide recommendations for future research directions.

1.8 Research Timeline

The research timeline can be seen in Table 1.1 facilitates organization, establishes deadlines, and enables progress monitoring.

Table 1.1 Research timeline.

Activity	**2024**	**2025**							
		Jan	**Feb**	**Mar**	**Apr**	**May**	**Jun**	**Jul**	**Aug**
Literature Review									
Create system model									
Writing Chapter I–III									
Simulation and scenario testing									
Evaluation and analysis									
Writing Chapter IV–V									
Writing paper for publication									
Submit paper for publication									
Paper review and acceptance									