Chat Funky: An Offline Bluetooth-Based Encrypted Chat Application for Disaster Communication

Mandar Joshi[1], Dr.Juita Raut[2]
[1]*Sonopant Dandekar Shikshan Mandal, Palghar (w)-401404, Maharashtra, Bharat*
[2]*Assitant Professor, Department of Information Technology S. D. S. M. College, Palghar, Maharashtra, Bharat*

***Abstract*—A solid way to stay in touch matters a lot these days. But when storms hit or networks go down, regular online methods usually stop working - leaving folks cut off from help and each other. Our project brings forward ChatFunky, an Android app that lets users exchange private messages directly through Bluetooth, no internet needed. It works completely offline, allowing close-by phones to send coded texts by linking via wireless signals instead. All messages stay on the device, locked down with AES-256 GCM encryption - powered by Android's Jetpack Security tools. Built to run smooth without slowing your phone, so it works even off-grid or after disasters hit. Tests across many Android phones checked how well ChatFunky handles steady links, sending texts, strong encryption, and saving battery life. This setup shows Bluetooth, when paired with up-to-date security methods, can give a solid private chat option during emergencies. Coming updates aim to add photo sharing, multi-person conversations, plus longer reach using Bluetooth mesh networks.**

***Index Terms*—Offline Communication, Bluetooth Chat, Disaster Response, Encrypted Messaging, Android Application, AES Encryption, Jetpack Security, Mesh Networking.**

I. INTRODUCTION

When disaster hits, staying in touch can make the difference between life and death, helping teams respond quickly while keeping track of what's happening on the ground. But regular systems - like phones and online connections - often break down when quakes, storms, or rising waters strike. Without electricity, with cell sites knocked out or traffic jamming up signals, standard gear tends to fail right when people depend on it.

To tackle these issues, more folks are turning toward standalone networks that don't rely on cell towers. These setups let gadgets link up straight to one another through close-range signals such as Bluetooth or Wi-Fi Direct instead. Especially Bluetooth - it runs low on power while still working across tons of devices for brief-distance data sharing.

This study introduces ChatFunky - a phone app using Bluetooth RFCOMM channels to let folks exchange messages securely on the fly, no web needed. Main idea? Build a tough messaging tool that keeps humans linked up even when cell towers go dark.

ChatFunky uses Android's Bluetooth feature along with built-in encryption tools so messages stay private and stored info stays safe. Messages travel straight from one phone to another - none of that data goes anywhere outside your device. It's easy to use, keeps things locked down tight, doesn't drain battery much, which works well out in remote spots, during local emergencies, or places where internet barely functions.

The rest of this paper breaks down like so: In Section II, we look at current tech plus past studies about offline comms setups. Moving to Section III, it dives into how things were built along with the overall design setup. Then in Section IV, you'll find specifics on actual build steps and choices made during development. When reaching Section V, data from trials shows up alongside checks on speed and efficiency. Later, Section VI points out what didn't work well while hinting at possible upgrades later. Finally, Section VII wraps everything up by summing key takeaways.

II. LITERATURE REVIEW

Off-grid messaging tools got more popular lately - especially when helps needed or privacy matters.

Giuseppe's team [1] built an e-voting setup using blockchain to keep records open and unchangeable, showing how decentralization boosts confidence in online exchanges. In the same way, M. Ali's group [2] introduced a vote-tracking method powered by blockchain, relying on smart contracts to handle checks automatically without revealing voter identity. Such work highlights why spreading control across networks helps cut reliance on single central hubs.

In wireless offline chat, Sharma's team [6] built a tool using Bluetooth plus Wi-Fi Direct for sending messages directly between phones - no internet needed. The setup proved stable links were possible among close-range Android gadgets through RFCOMM channels, tech that ChatFunky later leaned on for its core connection method.

Thomas alongside Fernando [7] looked into fine-tuning Bluetooth links to cut down delays while saving battery life. Findings showed it still works well for fast, light data exchange - especially with smart thread handling paired with clever cache use.

Some studies - like Kaur's team [8], along with Gupta's group [9] - looked into Bluetooth Mesh plus Zigbee when setting up temporary networks during crises; they found routing messages through multiple hops really extends how far signals can travel.

Even though things have improved, plenty of current apps skip encryption or depend on online services to handle info. But ChatFunky stands out - it locks everything right on your phone using Jetpack Security, so only you control your chats, nobody else can peek in.

III. METHODOLOGY

A. System Architecture

ChatFunky's layout builds on a split system - three layers stacked apart yet working together

User Interface Layer (Presentation Layer):

Handles how users engage, shows messages, also manages moving between screens. Built using XML designs that include Material elements.

Application Logic Layer (Service Layer):

Deals with Bluetooth tasks - finding devices, linking them, handling connections. Keeps threads running smoothly so messages move fast.

Data Management Layer:

Keeps messages safe plus stores them locally through Jetpack Security's Encrypted File setup. Every chat gets its own locked-up file.

This setup uses separate levels to keep things flexible, easy to expand, while also safeguarding information during transfer.

B. Wireless link setup using Bluetooth

Bluetooth works on the 2.4 GHz frequency used by many devices, sending small amounts of data without wires over brief distances; ChatFunky runs on older Bluetooth tech called RFCOMM that acts like a traditional wired serial link while allowing steady back-and-forth messaging.

The way people exchange messages happens like this:

The person starts scanning for gadgets.

Nearby gadgets show up - pick them by hand to link up.

A link through RFCOMM gets set up by picking a fixed ID beforehand.

Messages keep moving through input and output lines without stopping.

When it disconnects, the app shuts down data flows smoothly while refreshing the interface at the same time.

C. Scrambling plus how safeties handled

ChatFunky keeps your info safe using strong AES-257 GCM coding on saved messages. Each message locks up with a special MasterKey made by the Android Keystore setup. That way, just the app itself can unlock what's inside.

Security components include:

- AES-256-GCM: Handles message encryption when stored, also takes care of decryption later.
- Android KeyStore – keeps keys safe using built-in protection tools.
- SecureFile tool - handles locking files + keeps them safe once saved.
- Bluetooth's Link Layer keeps data safe while moving - using encryption built right in.

This mix of protections means if someone breaks into the device, messages still can't be pulled out of the app's space.

IV. IMPLEMENTATION DETAILS

ChatFunky runs on Android Studio Arctic Fox, built with Kotlin. It works from SDK 24 onward - so it fits most phones these days.

Key tools or libraries are:

- Jetpack Security for encryption
- RecyclerView handles messages that change often
- Glide makes working with images smoother while saving time and effort
- Material Components for interface design
- Kotlin Coroutines for asynchronous thread management

A. Crafting how things look on screen

The interface comprises:

- Device Picker Screen: Shows nearby Bluetooth gadgets you can connect to.
- Chat Window: Shows live messages along with time stamps.
- Chat History Screen: Opens up old talks that are locked tight.
- Every display stick to Google's Material Design rules so it's easy to read and use.

B. Coding Modules

- BluetoothChatService.kt handles sockets, also takes care of links plus data flow.
- SecureDataKeeper.kt: Takes care of saving plus pulling info safely.
- ChatWindowActivity.kt shows conversations while handling what users' type. It also reacts to entries made by people using it.
- DevicePickerActivity.kt handles scanning while managing pairings.
- The modular setup makes it simple to fix issues while also streamlining updates to features.

V. FINDINGS PLUS HOW THEY FIT TOGETHER

Tests were run in various conditions along with different device configs.

A. Functional Testing

All key functions - like finding devices, linking them, sending messages, while keeping data secure - passed testing without issues. Connection stayed solid every single time when within 10 meters.

B. Performance Testing

Bluetooth worked fine up to 10 meters; average reach was about 9.8 - connection stayed solid. Messages popped up after roughly 0.7 seconds, which feels quick enough. The CPU ran between 9% and 12%, no heavy load during long chats. Drained around 3–4% battery every half hour, so it's good for outdoor tasks The app used little power while constantly sending data through Bluetooth.

C. Security Testing

Encryption plus data saving got checked using several fake break-in tests. Outside entry tries didn't work, showing how tough the coding setup really is.

D. User Evaluation

Test users liked how easy and dependable the system felt. When running drills for emergencies, being able to work without internet helped teams stay in touch and organize on site

VI. LIMITATIONS

While ChatFunky hit what it aimed for, some limits still hang around - despite that, progress shows; yet certain hurdles don't vanish easily, even if gains exist

- Wireless signal works only within about 33 feet.
- Just handles text messages.
- Pairing by hand needed prior to talking.
- Just private talks - one person at a time, nothing shared with others.

VII. FUTURE SCOPE

Folks'll works on turning ChatFunky into a full-on offline chat hub down the line - building out features that link up seamlessly without needing net access.

- Multimedia and File Sharing:

Photos, videos, or docs can now be sent - just like on popular chat apps. Files shared this way stay locked down using the same encryption that protects messages.

- Bluetooth Mesh Networking:

A step-by-step relay method lets gadgets pass signals along, boosting reach past 10 meters. That way, notes can travel far by hopping between nearby units, creating a close-knit web of linked gear.

- Group and Broadcast Messaging:
- Backs several gadgets at once - ideal for group chats or squad planning.
- Emergency Alert System:

Tap once to send an SOS signal straight to devices around you.

- Cross-Platform Compatibility:

Building updates for iOS plus computers to make things easier to use.

- Integration with IoT:

Hooked up to sensors and trackers that send your position plus safety info when trouble hits.

VIII. CONCLUSION

The study shows offline Bluetooth chats – protected by encryption – work well during disasters where networks fail. ChatFunky hits its goal: sending texts without Wi-Fi or signal, using AES-256 to keep everything private.

Tests showed the setup runs smooth, uses almost no energy, while keeping steady close-proximity links. Its block-style layout allows room to grow later on – think sharing audio or connecting devices in a web-like structure.

ChatFunky is a solid move toward strong comms tech that works without fixed setups – ones helping people keep in touch even when every other option drops out.

REFERENCES

[1] G. R. P. J. M. J., along with G. Fiumera, "An e-voting method using blockchain," IEEE Access, no. 8, pages 101–108, published in 2020.

[2] M. Ali, along with I. Imran, Z. Shah, S. Khattak, and H. Arshad published a paper titled "Blockchain-Powered Secure Voting Systems" in the International Journal of Computer Applications; it appeared in volume 176, issue 2, pages 34 to 41 during 2021.

[3] Le Yu, H. Li, W. Xue - along with Q. Li - published “A Decentralized Voting System Combining Blockchain and Digital Signatures” in IEEE Trans. Dependable Systems; volume 17, issue 4 spans pages 567 to 576, released in 2020.

[4] Sarit Kumar, along with R. Bera, also R. Chandra, put together a voting setup using blockchain for more openness - published in International Journal of Secure Systems, volume 14, issue 3, pages 210 to 217, came out in 2021.

[5] Kannan M., along with S. Saravanan but also S. Karthik presented an Ethereum-driven secure voting setup at the IEEE Conference on Blockchain Tech, pages 65 through 71 during 2020.

[6] Sharma, alongside K. Verma, plus N. Singh explored building a messaging setup that works without internet by using Bluetooth tech - paper featured in International Journal of Mobile Computing and Networking back in 2020, volume 12, issue three, pages forty-five through fifty-two.

[7] R. Thomas with G. Fernando, “Building a Bluetooth-Powered Instant Messaging App,” IEEE Conf. Wireless Communication Systems, pages 89–95, 2021.

[8] L. Kaur, D. Patel, M. Roy – "Bluetooth Mesh Networks in Crisis Messaging," presented at Int. Conf. on Disaster Recovery Systems, pages 233 to 240, held in 2019.

[9] S. Gupta, along with T. Menon plus P. Rao, explored offline disaster messaging using Bluetooth combined with Wi-Fi Direct - study featured in IEEE Trans. Humanitarian Tech., volume 6, issue 2, pages 65 to 74, published 2021.

[10] Google Developers, “Jetpack Security Library,” [Online]. Accessible: https://developer.android.com/topic/security/data

[11] Android Devs, "All About Bluetooth," [Online]. Found at: https://developer.android.com/guide/topics/connectivity/bluetooth