

**SECURE LOCATION DIGISIGN: A
PROXIMITY-ENFORCED NFC
DIGITAL SIGNATURE
FRAMEWORK WITH
CERTIFICATE-BASED
AUTHENTICATION AND HYBRID
WI-FI DIRECT TRANSFER**

**Thesis Submitted
in Partial Fulfillment of the Requirements for the
Degree of**

MASTER OF TECHNOLOGY

**in
Software Engineering**

**by
Mudit Jain
(Roll No. 24/SWE/02)**

**Under the Supervision of
Dr. Divyashikha Sethia**
Associate Professor
Department of Software Engineering
Delhi Technological University



**Department of Software Engineering
DELHI TECHNOLOGICAL UNIVERSITY**

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to **Dr. Divyashikha Sethia**, Associate Professor, Department of Software Engineering, Delhi Technological University, for her invaluable guidance, support, and encouragement throughout the preparation of this report. Her insights and expertise were instrumental in shaping the direction of my research.

I would like to thank all the teachers and students of the Software Engineering department at Delhi Technological University for providing continuous support and valuable suggestions. Lastly, I would like to thank all my family and friends for providing continuous support and motivation during the process of this study.

Mudit Jain

4th Semester

24/SWE/02

Department of Software Engineering

Delhi Technological University



DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-42

CANDIDATE'S DECLARATION

I Mudit Jain hereby certify that the work which is being presented in the thesis entitled SECURE LOCATION DIGISIGN: A PROXIMITY-ENFORCED NFC DIGITAL SIGNATURE FRAMEWORK WITH CERTIFICATE-BASED AUTHENTICATION AND HYBRID WI-FI DIRECT TRANSFER in partial fulfillment of the requirements for the award of the Degree of Master of Technology, submitted in the Department of Software Engineering, Delhi Technological University is an authentic record of my own work carried out during the period from 2024 to 2026 under the supervision of Dr Divyashika Sethia.

The matter presented in the thesis has not been submitted by me for the award of any other degree of this or any other Institute.

Candidate's Signature

This is to certify that the student has incorporated all the corrections suggested by the examiners in the thesis and the statement made by the candidate is correct to the best of our knowledge.

Signature of Supervisor (s)

**Signature of External
Examiner**



DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daultapur, Main Bawana Road, Delhi-42

CERTIFICATE BY THE SUPERVISOR(s)

Certified that **Mudit Jain** (enrollment no. 24/SWE/02) has carried out their search work presented in this thesis entitled “**Secure Location DigiSign: A Proximity-Enforced NFC Digital Signature Framework with Certificate-Based Authentication and Hybrid Wi-Fi Direct Transfer**” for the award of **Master of Technology** from Department of Software Engineering, Delhi Technological University, Delhi, under my supervision. The thesis embodies results of original work, and studies are carried out by the student himself and the contents of the thesis do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Signature

Dr. Divyashikha Sethia

Associate Professor

Department of Software Engineering

Delhi Technological University

Date:_____

Secure Location DigiSign: A Proximity-Enforced Nfc Digital Signature Framework With Certificate-Based Authentication and Hybrid Wi-Fi Direct Transfer

Mudit Jain

ABSTRACT

The increasing reliance on digital processes in business, medical, and legal settings calls for the implementation of methods for securely signing documents. The existing methods using remote document signing techniques have several weaknesses, which include man-in-the-middle attacks, interception, and vulnerability to network issues, but do not offer cryptographic proof of presence at the signing session.

This thesis focuses on the design and experimental analysis of two innovative digital signature systems based on the concept of proximity, implemented successively. The first of these systems, termed the **Secure Location DigiSign (SLDS) system**, is presented as a new certificate-based NFC digital signature system. This allows for interaction between the Signee mobile application working in the HCE mode and the Signer PC application connected to a PC/SC reader through an NFC connection. The SLDS system makes use of RSA-2048 credentials and X.509 certificates provided by a CA server and embedded into tamper-proof Secure Elements residing in each of the two devices. The process of mutual authentication occurs using a mutual challenge-response procedure, followed by the encrypted communication of the documents and their RSA-SHA256 signing using the NFC link. The security of the SLDS mutual authentication protocol was rigorously verified using the Scyther security tool for protocols assuming the Dolev-Yao adversary model, demonstrating its robustness against man-in-the-middle, replay, and impersonation attacks even when the number of sessions is unbounded. SLDS takes 253.5 seconds to sign a 500 KB document during one signing session, with cryptographic functions taking up 13.5 seconds of this time.

However, the basic bandwidth limitations of NFC, which amount to 106–424 kbps, make pure NFC-based document transfer impractical, at least for realistic file sizes – a 5 MB document, for instance, requires more than 60 minutes. The proposed **Secure Location DigiSign Enhanced (SLDS-E)** framework, a hybrid proximity digital signing system that designates NFC exclusively for proximity-enforced mutual authentication and Wi-Fi Direct credential exchange, and delegates any and all document transfer to a high-speed Transmission Control Protocol (TCP) channel over Wi-Fi Direct secured with Advanced Encryption Standard 256-bit Galois/Counter

Mode (AES-256-GCM), utilizing the NFC-negotiated session key to enforce an end-to-end cryptographic continuity across both channels alleviates this critical bottleneck. Wi-Fi Direct credentials are delivered exclusively over the authenticated NFC channel, ensuring that physical co-location remains a mandatory prerequisite for initiating any document transfer. Experimental evaluation demonstrates throughput improvements of $40\times$ to $720\times$ over pure NFC transfer across document sizes from 200 KB to 5 MB, reducing a 500 KB signing session from 253.5 seconds to 10 seconds, a 96% reduction while preserving all security guarantees of the original SLDS framework including mutual authentication, proximity enforcement, and cryptographic non-repudiation.

Both frameworks are evaluated against enterprise-grade security requirements and validated through a healthcare insurance prescription signing use case. The results confirm that SLDS-E constitutes a practical, secure, and high-performance solution for proximity-based digital document signing in professional environments.

Keywords: Near Field Communication, Digital Signatures, Host Card Emulation, PC/SC, Certificate Authority, RSA-2048, X.509, Mutual Authentication, Wi-Fi Direct, AES-256-GCM, Proximity Security, Secure Element, Scyther Formal Verification.

LIST OF PUBLICATIONS

The following research papers have been accepted for publication based on the work carried out in this dissertation:

1. **Submission ID: 1453**

Title: “Secure Location DigiSign - A Certificate-Based NFC Digital Signature System with PC Signer and Mobile-Based Document Signee”

Conference: 7th International Conference on Data Analytics & Management (ICDAM-2026)

Organized By: London Metropolitan University (UK)

In association with WSG University Portalegre Polytechnic University, SGGW Management Institute

Date: 12–14 June 2026

2. **Submission ID: 1429**

Title: “Secure Location DigiSign with Wi-Fi Direct - A Hybrid NFC Authentication Application for High-Speed Proximity Digital Signing”

Conference: 7th International Conference on Data Analytics & Management (ICDAM-2026)

Organized By: London Metropolitan University (UK)

In association with WSG University Portalegre Polytechnic University, SGGW Management Institute

Date: 12–14 June 2026

TABLE OF CONTENTS

Contents

Acknowledgements	1
Candidate's Declaration	1
Certificate	1
Abstract	1
List Of Publications	3
List Of Tables	iv
1 Introduction	1
1.1 Overview	1
1.2 Motivation	5
1.3 Problem Statement	8
1.4 Proposed Solution	9
1.5 Main Contributions of the Thesis	10
1.6 Thesis Layout	11
2 Technical Background	13
2.1 Near Field Communication (NFC)	13
2.2 Host Card Emulation (HCE)	15
2.3 PC/SC Readers	16
2.4 Certificate Authority and X.509 Certificates	16
2.5 RSA-2048 Cryptography	17
2.6 Secure Elements	18
2.7 SHA-256 Hash Function	18
2.8 AES-256-GCM Authenticated Encryption	19
2.9 Wi-Fi Direct	20
2.10 Scyther Formal Verification Tool	21

3	Proposed SLDS Framework	23
3.1	Introduction	23
3.2	Related Work	24
3.2.1	Overview of Existing Digital Signing Solutions	24
3.2.2	Research Gaps	25
3.3	System Architecture	26
3.3.1	System Components	26
3.4	Device Registration	27
3.5	Mutual Authentication Protocol	29
3.6	Secure File Transfer	31
3.7	Digital Signing and Verification	33
3.8	Novelty of SLDS	34
3.9	Security Analysis	35
3.10	Formal Verification Using Scyther	37
3.10.1	SPDL Model	37
3.10.2	Verification Results	38
3.11	Performance Analysis	39
3.12	Use Case: Healthcare Insurance Prescription Signing	40
4	Proposed SLDS-E Application	41
4.1	Introduction	41
4.2	Related Work and Research Gaps	42
4.2.1	Limitations of Existing Proximity Signing Frameworks	42
4.2.2	Research Gaps Addressed by SLDS-E	43
4.3	System Architecture	43
4.3.1	System Components	45
4.4	Phase 1: NFC Mutual Authentication	46
4.5	Phase 2: Wi-Fi Direct Credential Exchange	46
4.6	Phase 3: Wi-Fi Direct Network Association	47
4.7	Phase 4: AES-256-GCM Secured Document Transfer	47
4.8	Novelty of SLDS-E	49
4.9	Security Analysis	50
4.10	Performance Evaluation	51
4.10.1	Resource Efficiency	51
4.10.2	Transfer Throughput Comparison	52
4.10.3	End-to-End Signing Latency	53
4.11	Application Screenshots	54
4.12	Use Case: Healthcare Insurance Prescription Signing	55
5	Results and Performance Analysis	57
5.1	Introduction	57

5.2	Cryptographic Resource Comparison	57
5.3	Transfer Throughput Analysis	58
5.4	End-to-End Signing Latency	60
5.5	Security Overhead Analysis	61
5.6	Qualitative Security Comparison	62
5.7	Discussion	62
6	Conclusion, Future Scope and Social Impact	64
6.1	Conclusion	64
6.2	Limitations	66
6.3	Future Work	67
6.3.1	Social Impact	69
	Appendix I	74
I.1	Paper Acceptance Proofs	76
I.2	Scopus Indexing Proof	78
I.3	Conference Paper Registration Receipts	79
	Author's Profile	81

LIST of TABLES

3.1	Comparison of Existing Digital Signature Solutions With The Proposed SLDS Framework.	25
3.2	Cryptographic Resource Sizes For The SLDS Framework. . . .	28
3.3	Notations For The SLDS Mutual Authentication Protocol. . .	29
3.4	SLDS End-to-End Timing Analysis For A 500 KB PDF Document.	39
4.1	Additional Notations For The SLDS-E Framework Beyond Those defined For SLDS.	45
4.2	Cryptographic Resource Sizes For SLDS-E Per Device.	52
4.3	Transfer Time Comparison Between SLDS (NFC) And SLDS-E (Wi-Fi Direct) Across Document Sizes.	52
4.4	SLDS-E End-to-End Timing For A 500 KB PDF Signing Session. .	53
5.1	Cryptographic Resource Comparison Between SLDS And SLDS-E Per Device.	58
5.2	Transfer Time And Speedup Comparison Between SLDS And SLDS-E Across Document Sizes.	58
5.3	Phase-By-Phase Timing Comparison For A 500 KB PDF Signing Session In SLDS And SLDS-E.	60
5.4	Constant Security Overhead Comparison Between SLDS And SLDS-E, Independent Of Document Size.	61
5.5	Qualitative Security And Functionality Comparison Of SLDS and SLDS-E Against Existing Solutions.	62

LIST of FIGURES

1.1	Digital signature generation and verification process. The signer computes a hash of the document and encrypts it with their private key to produce the signature. The verifier independently recomputes the hash and decrypts the signature using the signer's public key to confirm authenticity and integrity.	2
1.2	Comparison of remote signing architecture versus proximity-based NFC signing. Remote architectures expose the signing channel to man-in-the-middle interception and replay attacks over the network. NFC-based proximity signing physically enforces co-location, eliminating the remote attack surface and providing implicit proof of locality.	3
1.3	NFC transfer time vs. document size	4
1.4	High-level overview of SLDS and SLDS-E framework contributions.	5
1.5	Limitations of existing signing solutions and proposed remedies.	7
2.1	The three operating modes of NFC.	14
2.2	Hardware Secure Element vs. Host Card Emulation architecture.	15
2.3	AES-256-GCM packet structure used in SLDS-E over the Wi-Fi Direct channel.	19
3.1	SLDS framework system architecture.	26
3.2	SLDS mutual authentication protocol sequence	30
3.3	SLDS framework flow diagram.	35
3.4	Scyther verification results for the SLDS mutual authentication protocol.	38
4.1	SLDS-E system architecture showing the two-channel hybrid communication model.	44
4.2	Complete SLDS-E four-phase protocol sequence.	49
4.3	SLDS-E Android Signee application interface.	54

4.4	RSA-SHA256 signature embedded in signed PDF metadata verified in Adobe Acrobat.	55
5.1	Transfer time comparison between SLDS and SLDS-E across document sizes from 200 KB to 5 MB.	59

Chapter 1

Introduction

1.1 Overview

The possibility to sign documents digitally and with certifiability has become an essential feature in almost all domains – healthcare, law, finance, enterprises, etc. Digital signatures cryptographically guarantee the authenticity, integrity, and non-repudiation of documents, and serve as the electronic equivalent of handwritten signatures, with much stronger security properties [1]. As organizations further migrate from paper-based to non-paper dependent workflows, the need for reliable digital signing solutions in various environments has grown tremendously. A digital signature is created by processing the document content through a cryptographic hash function and encrypting the resulting hash digest with the signer’s private key. Given the signatory’s public key, any recipient can subjectively verify the signatures by decrypting it and matching the results against another hash of the received document. These security properties are simultaneously guaranteed using digital signatures, which requires authentication, which confirms the identity of the signing document, integrity, which guarantees that the document has not been modified since its signing, and non-repudiation, which prevents the document signer from denying he signed the document [1]. Such a security combination is critical when documents have legal, financial, or medical implications. The general process of digital signature generation and verification is shown in Figure 1.1.

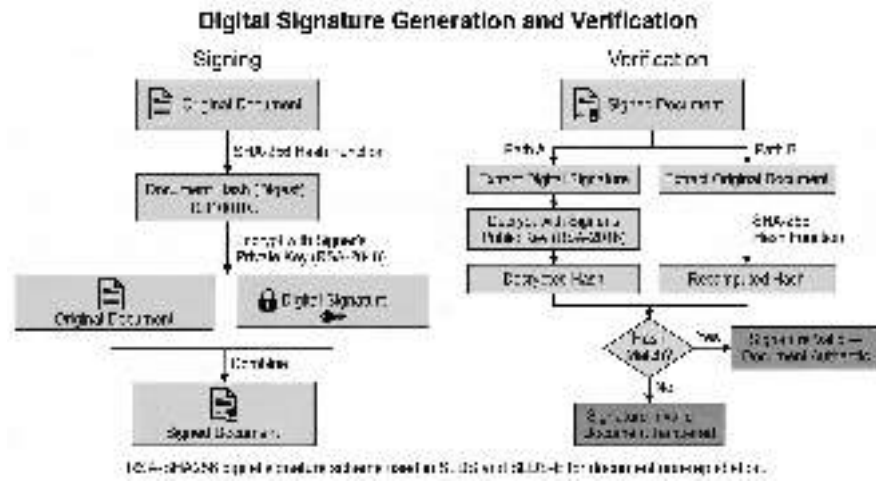


Figure 1.1: Digital signature generation and verification process. The signer computes a hash of the document and encrypts it with their private key to produce the signature. The verifier independently recomputes the hash and decrypts the signature using the signer’s public key to confirm authenticity and integrity.

The most common form of digital signature systems is centralized, server-based architectures, and signing takes place over networks connected to remote infrastructure. These approaches are functional in stable, connected environments, but they pose a variety of major security risks. Remote signing architectures are inherently susceptible to man-in-the-middle attacks, unauthorized interception, and replay attacks, where adversaries positioned between the parties may intercept, modify, or replay signing requests [2]. In addition, a cloud-dependent system introduces data privacy concerns, compliance issues, and challenges on network-challenged environments. The dependence on continual internet connections and working remote servers means these systems cannot be used in places like field hospitals, legal offices in remote locations, or anywhere else without guaranteed network reliability [3]. A fundamental limitation shared by all remote signing architectures is the absence of cryptographic proof of physical co-location between the document sender and the signing party. In many professional contexts, particularly in healthcare prescription signing, legal contract execution, and financial instrument authorization, the physical presence of both parties at the point of signing constitutes a legal and procedural requirement that cannot be satisfied by a network-mediated transaction. Figure 1.2 contrasts the remote signing model with the proximity-based model proposed in this thesis, illustrating how NFC’s physical constraint eliminates the attack surface inherent in remote architectures.

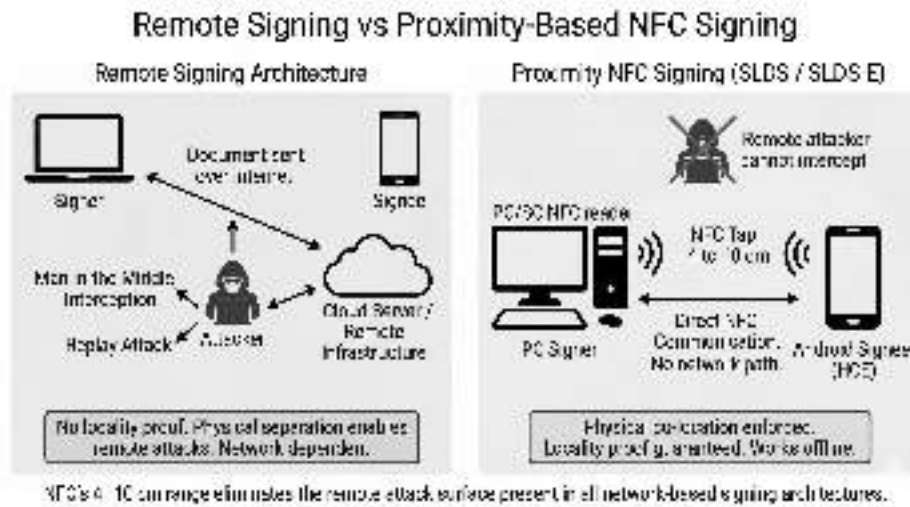


Figure 1.2: Comparison of remote signing architecture versus proximity-based NFC signing. Remote architectures expose the signing channel to man-in-the-middle interception and replay attacks over the network. NFC-based proximity signing physically enforces co-location, eliminating the remote attack surface and providing implicit proof of locality.

As described above, NFC technology operates at 13.56 MHz frequency in the vicinity of 4 – 10 cm, hence introducing a proximity limitation on hardware level and resolving the problem mentioned above [4]. As the NFC-based communication requires the presence of communicating parties in close proximity to each other, NFC is not subject to attacks via remote interception. In addition, the physical requirement of being in close proximity allows for the implication of local presence without having to prove it through digital signatures. Due to a very short range in which the NFC operates, it is physically impossible to intercept NFC traffic due to the fact that any third party interested in doing so would need to be physically close (within few centimeters) to the communicating parties. Currently available Android mobile devices offer an opportunity for smart card emulation using HCE technique, which makes NFC signing possible without special hardware on mobile devices [5].

This PKI architecture that works on the basis of X.509 certificates serves as the proven cryptographic underpinning of device authentication in this system. The Certificate Authority (CA) acts as an independent third party that provides digitally signed certificates binding the identity of the device to its public key. Both devices have the ability to verify the identity of the other device using the certificate provided by the Certificate Authority without having to contact the Certificate Authority when they need to authenticate their identities. RSA-2048 key pairs, including private and public

keys, are stored in tamper-proof Secure Elements within each device and do not reveal the private keys to the application layer. Therefore, credential access is protected by the physical security mechanism and credentials cannot be extracted even if the device is compromised.

However, NFC's security advantages come at a significant cost: its maximum bandwidth of 106–424 kbps imposes a fundamental throughput bottleneck that renders pure NFC-based document transfer impractical for real-world file sizes. Experimental measurements confirm that a 500 KB document requires approximately 253.5 seconds for a complete NFC-based signing transaction, with 240 seconds consumed by bidirectional NFC file transfer alone. A 5 MB document, such as medical imaging records or multi-page legal contracts, needs more than 60 minutes to transfer. This makes it impractical for use in professional settings. Figure 1.3 shows how NFC transfer time increases with document size. It highlights the seriousness of this issue for typical document sizes.

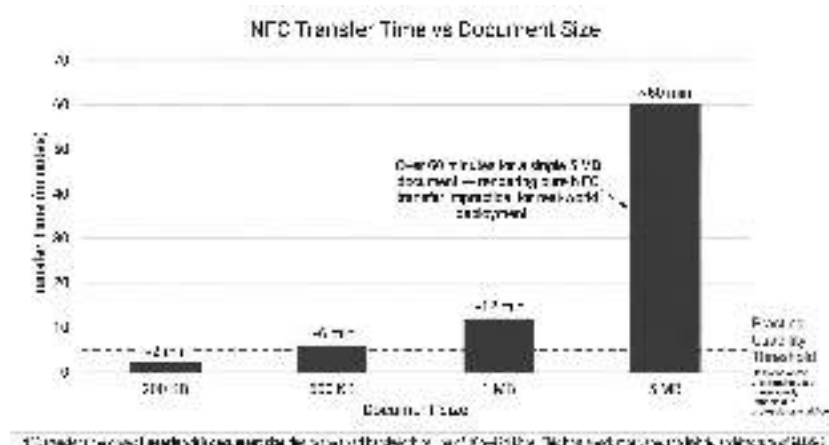


Figure 1.3: NFC transfer time vs. document size

The certified Wi-Fi Direct technology allows wireless communication among devices. This form of communication allows data exchange several magnitudes faster than that enabled by NFC and does not require any external access point nor Internet connection.

Security of the connection within a Wi-Fi Direct group is established using the WPA2 security protocol. Within such a group, a particular device will act as the software access point, while the rest of the devices will behave as regular IEEE 802.11 devices.

Experimental studies have shown that the average throughput rate achieved by Wi-Fi Direct is 2.72 MBps, which is far larger compared to NFC, whose

maximum throughput rate amounts to only 0.05 MBps, making for a more than two magnitudes difference in performance.

Combining proximity authentication provided by NFC and fast data transfer rates provided by Wi-Fi Direct for the purpose of exchanging documents would result in a hybrid architecture that takes into account all three aforementioned aspects at once.

This thesis addresses both the security limitations of remote signing architectures and the throughput limitations of pure NFC transfer through the design, implementation, and experimental evaluation of two complementary frameworks: Secure Location DigiSign (SLDS) and Secure Location DigiSign Enhanced (SLDS-E). Figure 1.4 presents a high-level overview of both frameworks and their relationship, illustrating how SLDS-E evolves from SLDS by introducing the Wi-Fi Direct transfer channel while retaining the complete NFC authentication foundation.

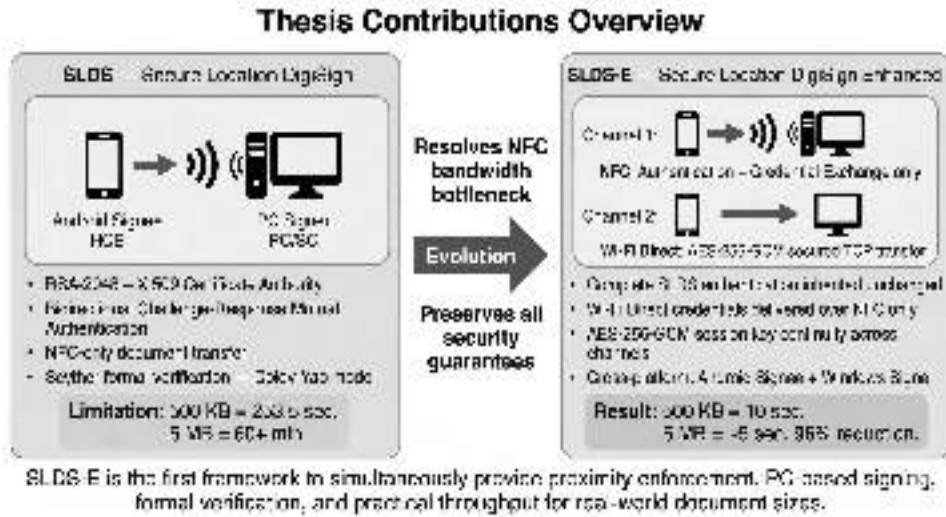


Figure 1.4: High-level overview of SLDS and SLDS-E framework contributions.

1.2 Motivation

The research undertaken as part of the dissertation is prompted by the coincidence of three major trends characteristic of document processing in companies and healthcare organizations. Each of these trends, in isolation from one another, would justify the need for proximity-aware signing processes. Together, however, they specify a concrete problem domain which cannot be addressed by current solutions.

Firstly, an ongoing trend involves increasing automation of document workflows in regulated industries. An illustrative example can be provided by healthcare organizations, which are expected to store patient information in a digital format, including prescription medications, laboratory reports, and treatment authorizations that have been digitally signed. In addition, law firms are expected to sign documents using electronic signatures and store the same in a digital format. Banks also are increasingly required to maintain accurate transaction records backed up by the signatures which provide strong evidence of non-repudiation. In all cases, a signature carries great legal significance and can be challenged in court even several decades after signing. Therefore, cryptographically secure signing solutions with a long period of validity and reliable key management should be employed to protect oneself against various types of attacks and ensure verifiable authenticity of signed documents.

The second trend that motivates the present study lies in the recognition of intrinsic drawbacks of current cloud-based signing solutions. There are essentially two kinds of digital signatures used today; both kinds have fundamental limitations which preclude their use in professional signing processes involving enforcement of proximity between a signer and a document recipient. Specifically, there are web- and cloud-based solutions, which include commercial services like DocuSign and Adobe Sign. Undoubtedly, such products bring great user convenience and access to a wide variety of features, but they require constant access to the Internet and rely on centralized servers as an execution environment. As a result, these digital signature tools are susceptible to a number of potential attacks, including man-in-the-middle attacks and server-side vulnerabilities. Crucially, no cloud-based solution has the capability of presenting a signer and a document keeper with the physical presence of a document together for the act of signing. This aspect is crucial in certain legal jurisdictions and, especially, practical situations (e.g., wards of hospitals, courtrooms, and banks) [2].

The second category comprises NFC-based proximity signing frameworks, including iDSign [6] and SecureXfer [7]. These systems correctly leverage NFC's proximity constraint to enforce co-location and eliminate remote attack surfaces. However, both frameworks are constrained to mobile-to-mobile communication scenarios and lack support for PC-based signing with large-screen document review, which is essential for professional workflows in which documents must be carefully examined before signing. A physician signing a multi-page prescription, a lawyer reviewing a contract, or a financial officer authorizing a transaction record all require the ability to read and review the complete document on a full-size display before committing a signature. Mobile screen sizes are inadequate for this purpose in professional contexts.

Furthermore, both existing NFC frameworks use NFC as the sole file transfer channel, which, as experimental measurements confirm in this thesis, makes them impractical for documents larger than a few hundred kilobytes.

The third trend motivating this work is the maturation of Wi-Fi Direct as a practical peer-to-peer communication standard available on commodity Android and Windows devices. Wi-Fi Direct enables direct high-throughput wireless communication without infrastructure, making it an ideal complement to NFC in a hybrid architecture where NFC handles proximity-enforced authentication and Wi-Fi Direct handles bulk data transfer. The key insight exploited in this thesis is that Wi-Fi Direct network credentials can be delivered exclusively over the authenticated NFC channel, binding the high-speed transfer session to the physical NFC tap event and preserving the locality guarantee even after the communication channel transitions from NFC to Wi-Fi Direct.

Figure 1.5 summarizes the key gaps in existing solutions and how the proposed frameworks address each gap, providing a visual motivation for the hybrid architecture developed in this thesis.

Research Gaps and How SIDS and SIDS-F Address Them

Research Gap	Existing Solutions	How SIDS / SIDS-F Address It
No Locality Proof	✗ Remote signing platforms (DocuSign, Adobe Sign) require a physical signature device to sign documents remotely.	✓ SIDS / SIDS-F are mobile devices that can sign documents locally over Wi-Fi Direct, ensuring physical co-location and enabling NFC channel.
No PC-Based Signing	✗ Enterprise and Government workflows require large-screen document review and signing.	✓ SIDS / SIDS-F are mobile devices that can sign documents locally over Wi-Fi Direct, ensuring physical co-location and enabling NFC channel.
NFC Bandwidth Overhead	✗ Existing NFC frameworks (e.g., NFC Forum) transfer 5 MB documents over NFC channels.	✓ SIDS / SIDS-F leverage Wi-Fi Direct for bulk document transfer (up to 100 MB) and use NFC for authentication and channel establishment.
No Formal Security Verification	✗ Existing NFC frameworks (e.g., NFC Forum) do not provide formal security verification.	✓ SIDS / SIDS-F use formal security verification (e.g., DocuSign, Adobe Sign) to ensure document integrity and authenticity.

SIDS and SIDS-F are mobile devices that can sign documents locally over Wi-Fi Direct, ensuring physical co-location and enabling NFC channel.

Figure 1.5: Limitations of existing signing solutions and proposed remedies.

The motivation for this thesis therefore arises from three converging requirements that no existing solution simultaneously addresses. First, enterprise and healthcare workflows require a signing system that provides cryptographic proof of physical co-location, ruling out all remote architectures. Second, professional signers require large-screen document review

on workstation-class hardware before committing a signature, ruling out mobile-only solutions. Third, real-world documents routinely exceed several megabytes, requiring transfer throughput that NFC cannot provide, ruling out pure NFC transfer solutions. These requirements collectively motivate the hybrid, cross-platform, proximity-enforced signing architecture proposed and evaluated in this thesis.

1.3 Problem Statement

Current digital signature solutions present several critical limitations when considered for deployment in proximity-enforced professional environments. These limitations can be grouped into three categories: architectural vulnerabilities in remote signing systems, operational infeasibility of existing NFC-based systems, and the absence of a cross-platform hybrid solution.

Vulnerabilities in Remote Signing Architectures: Remote signing systems, whether web-based or cloud-dependent, are vulnerable to network-layer attacks. In a situation where Signer and Signee are in different locations, an attacker could eavesdrop and add malicious data to the signature request, replay earlier signing sessions, etc [8]. In a PKI that relies on certificates, the PKI system is inherently secure but can still be vulnerable through certificate forgery, delayed revocation, and man-in-the-middle attacks where certificate validation is performed across untrusted networks. PKI systems also depend on remote servers that must remain available at all times; as such they form a single point of failure that could render signing unavailable during important instances. Most importantly no remote solution can ensure that the person signing was actually there when the document was signed, which is an essential requirement for law, healthcare and financial transactions.

Practical Infeasibility of Existing NFC-Based systems: The available NFC-based proximity signing systems do alleviate the location problem but face two mutually dependent and fundamental drawbacks for practical usage. Firstly, systems like iDSign [6] and SecureXfer [7] only support mobile-to-mobile interaction, i.e. Workstations cannot be used to sign. This severely limits their applicability in the business sector where most users access applications from a desktop or laptop. Secondly, all available NFC-based systems rely solely on NFC for data transmission. NFC operates with bandwidth of between 106 and 424kbps. This means data transfer is slower depending on file size, and has been reported that a 500KB document took approximately 240 seconds for a bi-directional NFC transfer while 5MB took in excess of 60 mins [?]. These problems render the systems almost unusable given the high-resolution graphics, lengthy reports, and multi-page documents that are

commonplace in business documents. **Absence of Cross-Platform Hybrid Solutions:** Existing hybrid frameworks that decouple authentication from file transfer, such as SecureXfer which uses NFC for authentication and Bluetooth for file transfer, do not extend to PC-based signing and are constrained to mobile-to-mobile scenarios. Furthermore, they do not reuse the authentication session key for encrypting the transfer channel, creating a cryptographic discontinuity between the authentication and transfer phases that weakens the overall security model. No existing framework simultaneously provides PC-based signing with large-screen document review, NFC-enforced proximity authentication, end-to-end session key continuity, and high-speed wireless file transfer across heterogeneous operating systems.

The core problem addressed by this thesis is therefore the design of a proximity-based digital signing framework that simultaneously provides cryptographic proof of locality, supports PC-based document review and signing, achieves practical document transfer throughput for real-world file sizes, and maintains end-to-end cryptographic continuity across all communication phases.

1.4 Proposed Solution

This thesis proposes and evaluates two frameworks developed in progression to address the identified limitations comprehensively.

Secure Location DigiSign (SLDS) is proposed as a novel certificate-based NFC digital signature framework enabling proximity-enforced document signing between a mobile-based Signee application, operating in Host Card Emulation (HCE) mode, and a PC-based Signer application connected to a PC/SC NFC reader. Both devices obtain RSA-2048 key pairs and X.509 certificates from a Certificate Authority (CA) server and store credentials in tamper-resistant Secure Elements. This protocol sets up a mutual authentication mechanism, whereby there is a challenge-response on both directions of the protocol; where both the device's send 32 bytes random challenge that are individually encrypted using the respective private keys and their peers' response will be authenticated using the respective CA issued certificates. Upon mutual authentication, a symmetric session key will be established where all further communications are encrypted. The Signee would then pass the document using the NFC channel in encrypted fragments where the Signer applies an RSA-SHA256 digital signature, and send the signed document back. The mutual authentication protocol is formally verified using the Scyther tool under the Dolev-Yao adversary model, confirming resistance to man-in-the-middle, replay, and impersonation attacks across unbounded

sessions. SLDS addresses the locality, PC-based signing, and formal security verification requirements but retains NFC as the sole transfer channel, establishing the security foundation upon which SLDS-E is built.

Secure Location DigiSign Enhanced (SLDS-E) is proposed to resolve the NFC bandwidth bottleneck of SLDS while preserving all its security guarantees without modification. SLDS-E operates in four sequential phases. In the first phase, the complete SLDS mutual authentication protocol executes over NFC, establishing mutual trust and a shared AES-256 session key. In the second phase, the Signee delivers Wi-Fi Direct network credentials, comprising the network SSID, WPA2 passphrase, and TCP port number, to the Signer over the authenticated NFC channel. In the third phase, the PC Signer joins the Android-hosted Wi-Fi Direct group as a standard WPA2 IEEE 802.11 infrastructure client using the Windows Native WiFi API. In the fourth phase, all document data is transferred over a TCP connection secured with AES-256-GCM authenticated encryption using the session key established during NFC mutual authentication.

Since Wi-Fi Direct credentials are delivered exclusively over the authenticated NFC channel, physical co-location at the NFC tap remains a mandatory prerequisite for joining the Wi-Fi Direct network, preserving the locality-of-proof guarantee of SLDS. Experimental evaluation demonstrates throughput improvements of $40\times$ to $720\times$ over pure NFC transfer across document sizes from 200 KB to 5 MB, reducing a 500 KB signing session from 253.5 seconds to 10 seconds, a 96% reduction in end-to-end latency.

1.5 Main Contributions of the Thesis

The primary contributions of this thesis are as follows:

1. **Design and Implementation of the SLDS Application:** A novel NFC-based digital signature framework enabling a mobile Signee in HCE mode to tap a PC/SC reader connected to a PC Signer, establishing mutual certificate-based authentication, encrypted document transfer over NFC, and RSA-SHA256 digital signing, with RSA-2048 credentials and X.509 certificates stored in tamper-resistant Secure Elements on both devices.

A complete Scyther-based formal verification of the SLDS mutual authentication protocol modelled in Security Protocol Description Language (SPDL) under the Dolev-Yao adversary model, confirming session key secrecy, aliveness, non-injective synchronization, nonce commitment, and weak agreement for both Signer and Signee roles across

unbounded sessions.

2. **Design and Implementation of the SLDS-E Hybrid Application:** A hybrid communication architecture extending SLDS by retaining NFC for proximity-enforced mutual authentication while delegating bulk document transfer to an AES-256-GCM secured TCP channel over Wi-Fi Direct, with Wi-Fi Direct access credentials exchanged exclusively over the authenticated NFC channel to preserve the locality-of-proof guarantee across the channel transition.
3. **Quantitative Performance Evaluation:** Experimental benchmarking of SLDS-E across document sizes from 200 KB to 5 MB, demonstrating throughput improvements of $40\times$ to $720\times$ over pure NFC transfer and reducing a 500 KB signing session from 253.5 seconds to 10 seconds, a 96% reduction in end-to-end signing latency.
4. **Security Continuity Analysis:** Formal demonstration that transitioning bulk transfer from NFC to Wi-Fi Direct introduces no new attack surfaces, with dual-layer WPA2 and AES-256-GCM protection ensuring that compromise of the wireless link layer alone cannot recover plaintext document data, and in-band NFC credential delivery maintaining physical co-location as a mandatory prerequisite for initiating any transfer session.
5. **Healthcare Use Case Validation:** Application of both frameworks to the healthcare insurance prescription signing use case, demonstrating how SLDS-E resolves the operational infeasibility of SLDS for time-critical medical workflows involving multi-megabyte patient records.

1.6 Thesis Layout

The remainder of this thesis is organized as follows. Chapter 2 presents the technical background covering Near Field Communication, Host Card Emulation, PC/SC readers, Certificate Authority infrastructure and X.509 certificates, RSA-2048 cryptography, Secure Elements, Wi-Fi Direct, AES-256-GCM authenticated encryption, and the Scyther formal verification tool and Dolev-Yao adversary model. Chapter 3 presents the proposed SLDS framework, including a review of related work and identified research gaps, the complete system architecture and component description, the mutual authentication and file transfer protocols, the novelty contributions, security analysis, and formal Scyther verification results. Chapter 4 presents

the proposed SLDS-E framework, detailing the motivation arising from NFC bandwidth limitations measured in SLDS, the hybrid four-phase communication protocol, novelty contributions, and security continuity analysis confirming preservation of all SLDS security guarantees. Chapter 5 presents the combined results and performance analysis of both frameworks, including resource efficiency characteristics, transfer throughput benchmarking, end-to-end signing latency comparison, and a comprehensive comparative evaluation of SLDS versus SLDS-E. Chapter 6 concludes the thesis with a summary of key findings, limitations of the current implementations, and directions for future work including blockchain audit trail integration and multi-party proximity signing support.

Chapter 2

Technical Background

This chapter presents the foundational technologies and cryptographic primitives upon which the SLDS and SLDS-E frameworks are built. In every chapter there is enough theory and technical details to understand the design choices made in the coming chapters.

2.1 Near Field Communication (NFC)

Near Field Communication (NFC) is a short-range wireless communication technology operating at a carrier frequency of 13.56 MHz within a maximum range of 4–10 cm [4]. Standardized under ISO/IEC 14443 and ISO/IEC 18092, NFC enables bidirectional data exchange at transfer rates of 106, 212, or 424 kbps between compatible devices [3]. NFC is an evolution of Radio Frequency Identification (RFID) technology, inheriting its contactless communication model while extending it with bidirectional peer-to-peer capability and standardized application-layer protocols.

NFC operates in three distinct modes, as illustrated in Figure 2.1. In Reader/Writer mode, an active NFC device reads data from or writes data to passive NFC tags, commonly used for product authentication and information retrieval. In Peer-to-Peer mode, two active NFC-enabled devices communicate bidirectionally to exchange data such as contacts or configuration parameters, following the ISO/IEC 18092 standard. In Card Emulation mode, an NFC-enabled device emulates a passive contactless smart card, allowing it to interact with existing NFC readers — this mode is the basis for Host Card Emulation discussed in Section 2.2.

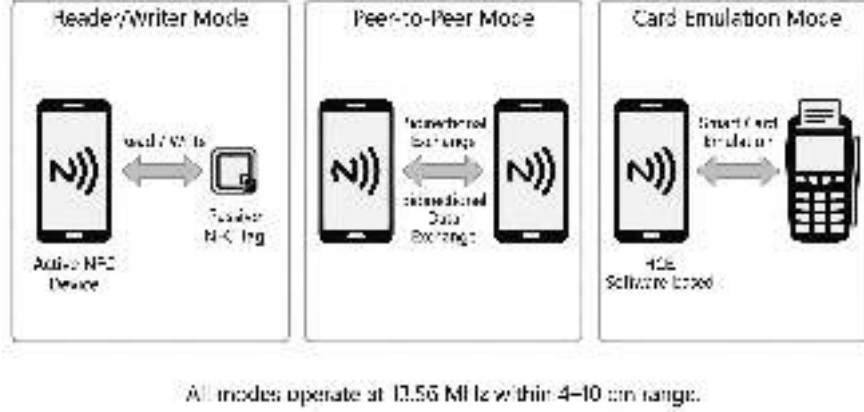


Figure 2.1: The three operating modes of NFC.

NFC's most significant security property is its extremely short operating range. Any adversary attempting to eavesdrop on or intercept an NFC communication must physically position themselves within centimetres of the communicating devices, making remote attacks physically infeasible under normal operating conditions [4]. This proximity constraint forms the foundational security property exploited by both SLDS and SLDS-E to enforce physical co-location between the Signer and Signee.

NFC communication follows the Application Protocol Data Unit (APDU) model defined in ISO/IEC 7816-4. Command APDUs are sent from the reader to the card-emulating device and consist of a Class byte (CLA), Instruction byte (INS), two Parameter bytes (P1, P2), an optional data field, and an expected response length field (Le). Response APDUs returned by the card-emulating device consist of an optional data field and two Status bytes (SW1, SW2) indicating success or error conditions.

A significant drawback of NFC with respect to this thesis is the upper bound of bandwidth. It has a peak transfer speed of 424 kbit/sec. Considering APDU framing overhead the actual throughput for a mass data transfer on NFC is somewhere about 40-50 KBytes/sec. It is a constraint of the hardware physical contactless interface using an inductive link and cannot be solved by software modification. This constraint has brought SLDS-E to a hybrid approach. In this structure, NFC is used only for authentication, while bulk document transfer is handled by Wi-Fi Direct.

2.2 Host Card Emulation (HCE)

Host Card Emulation (HCE) is a software structure that allows an Android device to imitate a contactless smart card without needing special hardware called a Secure Element, which is managed by a mobile network operator [5]. HCE was introduced in Android 4.4. It enables application developers to register NFC Application Identifiers (AIDs) with the Android NFC service. This service directs incoming APDU commands from NFC readers to the registered application running on the device's main processor. Figure 2.2 illustrates the HCE architecture and contrasts it with hardware Secure Element-based card emulation. In a hardware Secure Element configuration, the NFC controller routes incoming APDU commands directly to the dedicated SE chip, bypassing the host CPU entirely. In the HCE model, the NFC controller routes APDUs to the host CPU via the Android NFC service, which dispatches them to the registered HCE service class for software processing.

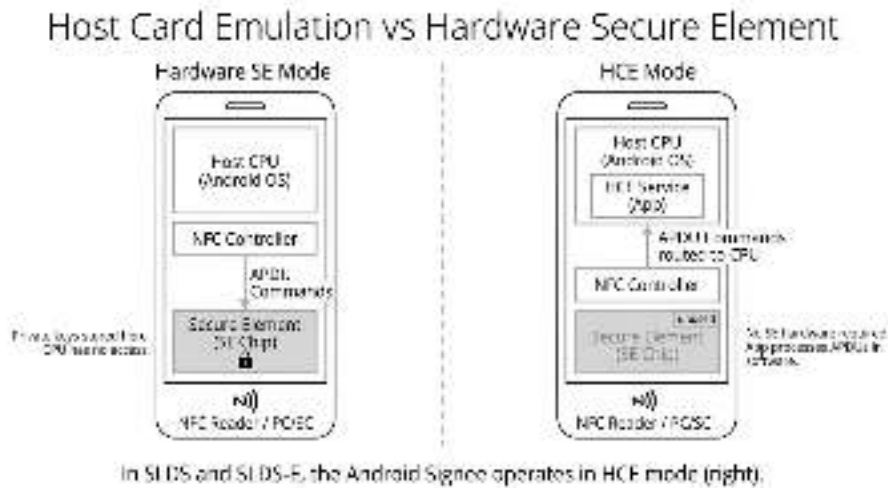


Figure 2.2: Hardware Secure Element vs. Host Card Emulation architecture.

HCE operation follows a clear sequence when the NFC reader makes contact. The external NFC reader sends a SELECT AID command to identify the target application. The Android NFC service matches the AID with registered HCE services and activates the right service. The HCE service then receives the following command APDUs, processes them based on application logic, and returns response APDUs. The session continues until the NFC field is removed or the reader ends communication.

In the SLDS and SLDS-E frameworks, the Android Signee application provides an HCE service that manages the full mutual authentication APDU exchange. The HCE model removes the need for special hardware. This means the framework can be used on any NFC-capable Android device that runs Android 4.4 or later, without the need for agreements with mobile operators or hardware changes.

2.3 PC/SC Readers

PC/SC (Personal Computer/Smart Card) is a standard interface specification that defines how personal computers communicate with smart card readers and NFC-capable devices [9]. PC/SC readers are USB-connected devices that follow ISO/IEC 14443. They allow desktop or laptop computers to communicate with Android devices running HCE services. The PC/SC framework offers standardized APIs, available on Windows through the Win-SCard library. These APIs enable desktop applications to send APDU commands to connected NFC readers and receive APDU responses. By providing this abstraction layer, desktop NFC applications do not have to care about the reader they are talking to. In the SLDS and SLDS-E frameworks, the PC Signer application uses the PC/SC API to initiate NFC communication upon card detection, transmit authentication APDU commands, and receive response APDUs. The experimental evaluation uses the SCL011 USB NFC reader, which is fully compliant with ISO/IEC 14443 and compatible with Android HCE services. A practical constraint for PC/SC communication is the APDU data length limitation of ISO/IEC 7816-4, which supports a maximum of 255 bytes per standard APDU data field. In SLDS, document transfer is therefore implemented using chunked transmission, dividing the encrypted file into sequential APDU-sized chunks.

2.4 Certificate Authority and X.509 Certificates

A certificate authority (CA) is a trusted third party that issues, manages, and revokes digital certificates to link a cryptographic public key to a confirmed device identity [10]. The X.509 standard specifies the structure of digital certificates used in PKI systems [11]. An X.509 certificate contains the subject identity, issuer identity, subject public key, validity period, serial number, and the CA's digital signature over all certificate fields computed using the CA's private key.

A verifier confirms a certificate's authenticity by verifying the CA's signature using the CA's public key. If the signature is valid and the certificate is within its validity period, the verifier can trust that the contained public key genuinely belongs to the identified subject. This trust model eliminates the need for real-time CA communication at the point of authentication, as the CA's signature on the certificate is self-contained and verifiable offline.

In the SLDS and SLDS-E frameworks, a Python-based CA server generates RSA-2048 key pairs and X.509 certificates for both devices during registration. Each device receives its own certificate, its private key, and the CA's certificate. During mutual authentication, each device presents its certificate to the peer, which verifies the CA's signature before proceeding with the challenge-response protocol. Both the devices are thus registered and verified to be registered to the same trusted CA before the document transfer takes place.

2.5 RSA-2048 Cryptography

RSA (Rivest–Shamir–Adleman) is a widely deployed asymmetric cryptographic algorithm whose security rests on the computational intractability of factoring the product of two large prime numbers [1]. RSA key generation selects two large primes p and q , computes the modulus $n = p \cdot q$, chooses a public exponent e (commonly 65537), and derives the private exponent d satisfying $e \cdot d \equiv 1 \pmod{(p-1)(q-1)}$.

For digital signature generation, the signer computes a hash of the document and raises it to the power d modulo n :

$$\sigma = H(m)^d \bmod n \quad (2.1)$$

For verification, the verifier raises the signature to the power e modulo n and compares the result to an independently computed hash:

$$H(m) \stackrel{?}{=} \sigma^e \bmod n \quad (2.2)$$

RSA-2048 uses a 2048-bit modulus, providing approximately 112 bits of security, which is the current NIST recommended minimum for long-term security. In the SLDS and SLDS-E frameworks, RSA-2048 serves three purposes: generating X.509 certificates at the CA server, performing mutual authentication through private key-based challenge encryption, and generating RSA-SHA256 digital signatures on document hashes. RSA public and private keys each require 256 bytes of storage, and X.509 certificates require

approximately 1024 bytes, giving a total credential storage requirement of 1664 bytes per device.

2.6 Secure Elements

A Secure Element (SE) is a tamper-resistant hardware component made for the safe storage of sensitive data and the performance of cryptographic tasks in an isolated setting. This setup protects it from the host processor and its operating system [12]. As a result, these components offer strong hardware resistance against physical attacks, side-channel analysis, and software exploitation. The value comes from the fact that only cryptographic processes occur within a secure processor, and so the private key material is never put out in the clear.

In both the SLDS and SLDS-E, each device’s RSA-2048 private key material and its X.509 certificates are held in its Secure Element. All cryptographic tasks that use these private keys, including encrypting authentication challenges and creating RSA-SHA256 digital signatures, take place entirely within the Secure Element. This design provides strong protection against relay attacks. Even if a relay proxy successfully captures NFC communications and forwards them to a remote device, it cannot create valid cryptographic responses. This limitation comes from not having access to the Secure Element-protected private key, which is linked to the registered device and is never shared over the network.

2.7 SHA-256 Hash Function

The Secure Hash Algorithm 256-bit (SHA-256) is a cryptographic hash function standardized by NIST in FIPS 180-4 that maps an input of arbitrary length to a fixed-length 256-bit output called a message digest. SHA-256 satisfies the three fundamental properties of a cryptographic hash function: pre-image resistance, making it infeasible to recover an input from its hash; second pre-image resistance, making it infeasible to find a different input producing the same hash as a given input; and collision resistance, making it infeasible to find any two distinct inputs that produce the same hash output.

In the SLDS and SLDS-E frameworks, SHA-256 serves two roles. First, it is the hash component of the RSA-SHA256 digital signature scheme, in which the document hash is encrypted using the Signer’s RSA private key to produce the digital signature. Second, SHA-256 computes the document hash transmitted as part of the file metadata during secure file transfer,

enabling the Signer to verify document integrity after reception.

2.8 AES-256-GCM Authenticated Encryption

Advanced Encryption Standard 256-bit Galois/Counter Mode (AES-256-GCM) is an authenticated encryption scheme providing simultaneous confidentiality, integrity, and authenticity [13]. AES-256-GCM combines the AES-256 block cipher in Counter Mode for encryption with the Galois Message Authentication Code (GMAC) for integrity verification, producing both ciphertext and a 128-bit authentication tag in a single pass.

Each packet in SLDS-E is structured as shown in Figure 2.3, consisting of a 12-byte random Initialization Vector (IV), the AES-256-GCM ciphertext, and a 16-byte authentication tag.

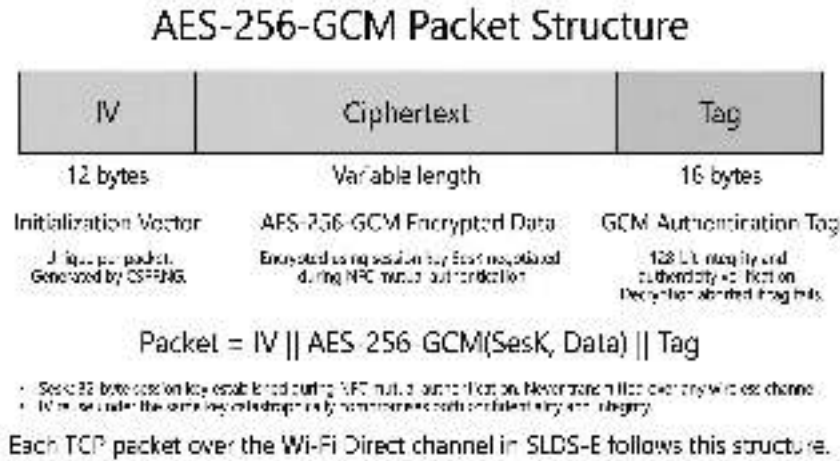


Figure 2.3: AES-256-GCM packet structure used in SLDS-E over the Wi-Fi Direct channel.

The decryption process verifies the 128-bit authentication tag before decrypting the ciphertext. If verification fails, decryption is aborted entirely, preventing any tampered or corrupted data from reaching the application. Each packet carries a unique 12-byte IV generated by a cryptographically secure random number generator, as IV reuse under the same key catastrophically compromises both confidentiality and integrity in GCM mode.

In SLDS-E, AES-256-GCM secures every TCP packet over the Wi-Fi Direct channel using the 32-byte session key *SesK* established during NFC mutual authentication. This reuse of the authentication session key for transfer encryption eliminates any cryptographic discontinuity between the au-

thentication and transfer phases, ensuring end-to-end protection across both communication channels.

2.9 Wi-Fi Direct

Wi-Fi Direct is a peer-to-peer wireless communication standard certified by the Wi-Fi Alliance. It allows devices to communicate directly with each other without needing an external access point or internet connection. It operates over the standard 2.4 GHz and 5 GHz IEEE 802.11 frequency bands. Wi-Fi Direct achieves throughput much higher than NFC while keeping the infrastructure-free operation that's essential for proximity signing workflows.

In a Wi-Fi Direct group, devices negotiate their roles during a Group Owner (GO) negotiation phase outlined in the Wi-Fi Peer-to-Peer (P2P) Technical Specification. The device that takes on the GO role acts as a software access point under WPA2-Personal security. It maintains a P2P group network with its own SSID and WPA2 passphrase, while other devices join as standard IEEE 802.11 infrastructure clients.

Importantly, older devices that support regular Wi-Fi but not full Wi-Fi Direct P2P negotiation can still join a Wi-Fi Direct group as ordinary WPA2 clients without needing specific drivers. This setup allows a Windows PC to connect to an Android-hosted Wi-Fi Direct group using the standard Windows Native WiFi API (WlanConnect), without any extra drivers or Wi-Fi Direct-specific software on the signer side. Wi-Fi Direct requires WPA2-Personal security for all group communications. It provides link-layer confidentiality through AES-CCMP encryption. The Group Owner establishes group credentials, which include the SSID and WPA2 passphrase. These credentials are shared with clients through a separate channel before they connect. Wi-Fi Direct works completely without internet access. This keeps the SLDS framework functional offline and removes the need for external network support. The WPA2 link-layer encryption offers a basic level of wireless confidentiality. It works independently and alongside the application-layer AES-256-GCM encryption used by SLDS-E. Together, these provide dual-layer protection for all document data during transfer. Experimental tests show an average Wi-Fi Direct TCP throughput of 2.72 MBps, with a 95th-percentile guaranteed throughput of 2.12 MBps. This is significantly better than NFC's typical limit of about 0.05 MBps. The throughput benefit exceeds 40 times at the lower end and can reach up to 720 times in real situations. This advantage is in direct support of the performance improvements of SLDS-E over SLDS.

In SLDS-E, prior to the NFC tap event, the Android Signee creates a

Wi-Fi Direct Group Owner network and a TCP server at port 8765. This avoids the Group Owner negotiation delay within the signing process critical path. Once both devices successfully authenticate through NFC, the Wi-Fi Direct SSID, WPA2 passphrase, and TCP port are conveyed to the PC Signee through the secure NFC channel as an encrypted credential package, C_{info} . This method of credential delivery explicitly binds the Wi-Fi Direct transfer session to the physical NFC tap event. Only a device that physically tapped the Android Signee within a proximity of 4 to 10 cm, and successfully underwent mutual certificate-based authentication would know of the required information to join the Wi-Fi Direct network and initiate the transfer of documents. This maintains the locality-of-proof assurance of the SLDS framework while transitioning from the NFC authentication channel to the higher-speed Wi-Fi Direct transfer channel.

2.10 Scyther Formal Verification Tool

Scyther is an automatic protocol security checker, developed by Cremers [14]. The tool uses symbolic analysis to check crypto protocols modeled with the Dolev-Yao adversary model. Input to Scyther is a protocol specification in SPDL language, and it systematically tries out all execution paths. This includes adversarial manipulations to verify or disprove security claims across unlimited concurrent sessions. Unlike model checkers that only work with limited state spaces, Scyther’s analysis of unbounded sessions offers better assurance that no attacks exist, no matter how many participants are involved. This makes it especially effective for checking authentication protocols used in open environments.

The Dolev-Yao model assumes that the adversary has complete control over the communication network and can intercept, modify, replay, and inject arbitrary messages, but cannot break computationally secure cryptographic primitives [15]. Under this model, Scyther verifies five classes of security claims: secrecy of specified terms, confirming that a value remains unknown to the adversary; aliveness, confirming that the peer participated in some run of the protocol; non-injective synchronization, ruling out man-in-the-middle modifications to exchanged messages; weak agreement on peer identity, ensuring both parties agree on who they are communicating with; and nonce commitment, preventing cross-session reuse of freshly generated random values.

In this thesis, Scyther is used to formally verify the SLDS mutual authentication protocol, modelling the complete eight-message exchange involving NFC initiation, certificate exchange, bidirectional challenge-response valida-

tion, and session key establishment across three roles Signer, Signee, and Certificate Authority. Scyther's confirmation that no attacks exist across unbounded concurrent sessions provides machine-checked assurance of protocol correctness, complementing the informal security analysis presented in Chapter 3 and establishing a formal foundation for the security claims of the SLDS-E framework.

Chapter 3

Proposed SLDS Framework

3.1 Introduction

The growing tendency to digitize business processes involving document exchanges within health care, legal, and financial sectors now demands signature solutions which simultaneously secure cryptographic integrity and enforce physical colocation of parties. Existing remote signing architectures introduce network-layer vulnerabilities and cannot guarantee that the signing party was physically present with the document at the time of signing. Existing NFC-based frameworks address the locality problem but are constrained to mobile-to-mobile scenarios and lack PC-based document review capabilities essential for professional workflows.

This chapter presents the Secure Location DigiSign (SLDS) framework, a novel certificate-based NFC digital signature system that enables a mobile-based Signee application, operating in Host Card Emulation (HCE) mode, to tap a PC/SC reader connected to a PC-based Signer application and digitally sign a document over the Near Field Communication (NFC) interface. SLDS employs RSA-2048 credentials and X.509 certificates issued by a Certificate Authority (CA) server, stored in tamper-resistant Secure Elements on both devices. The framework establishes mutual authentication through a bidirectional challenge-response protocol, followed by encrypted document transfer and RSA-SHA256 digital signing over the NFC channel.

The chapter is organized as follows. Section 3.2 reviews existing digital signature solutions and identifies the research gaps addressed by SLDS. Section 3.3 presents the system architecture and components. Sections 3.4 through 3.7 describe the three operational phases of SLDS in detail. Section 3.8 discusses the novelty contributions. Section 3.9 presents the security analysis. Section 3.10 presents formal verification results using the Scyther

tool. Section 3.12 presents the healthcare insurance use case.

3.2 Related Work

3.2.1 Overview of Existing Digital Signing Solutions

Various digital signing solutions have been proposed and deployed to facilitate secure document handling across enterprise environments. However, these systems frequently encounter significant limitations including slow processing times, susceptibility to man-in-the-middle attacks, dependence on centralized infrastructure, and inability to provide cryptographic proof of physical co-location. Legal, healthcare, and financial services in particular require in-person document signing with locality proof — a requirement that no existing remote solution can satisfy. The following analysis examines the most relevant existing solutions and their key limitations.

Ponnappalli and Saxena [16] proposed a browser-agnostic digital signature architecture supporting multiple standards including PKCS#7, CMS, and XML DSig, but the solution relied on Java applets and was unsuitable for mobile NFC environments. Weihua and Shuang [17] combined MD5 with RSA for a distributed rendering submission system, while Yulianto et al. [18] implemented three-tier authentication combining SHA-256, DSA, and 3DES. Both works focus on unidirectional server-side uploads rather than bidirectional proximity-enforced transfer. Kishore et al. [19] optimized RSA through parallel hashing using the OpenMP framework, achieving 4–6 times speedups for large files on multi-core processors, but targeted traditional desktop scenarios without any mobile NFC consideration. Fun et al. [13] investigated lightweight elliptic curve-based digital signature algorithms for resource-constrained environments, demonstrating that ECDSA achieves a favourable balance between security and computational efficiency, but without any proximity enforcement or file transfer mechanism.

Most relevant to the present work are iDSign [6] and SecureXfer [7]. Sharma and Sethia [6] proposed iDSign, employing Identity-Based Cryptography with a modified iTLS protocol and Secure Element storage, achieving a 97% reduction in cryptographic overhead compared to traditional certificate-based approaches. Jaiswal et al. [7] introduced SecureXfer, a hybrid mobile-based framework using NFC HCE for mutual RSA-based authentication and Bluetooth for file transfer. However, both frameworks are constrained to mobile-to-mobile communication, lacking PC-based signing with full-screen document review essential for professional workflows.

3.2.2 Research Gaps

The analysis of existing solutions reveals four critical gaps that the SLDS framework is designed to address:

1. **Lack of Proof of Locality:** Current remote solutions are vulnerable to network attacks and cannot provide cryptographic proof that the signing party was physically present with the document. There are still specific services that need the presence of the user, e.g., the signing of a legal contract or prescription, for which we do not know if the locality can be proved by cloud services.
2. **Absence of PC-Based Signing:** No NFC-proximity system provides large screen document review login in PC workstation. All current NFC solutions are constrained to mobile-to-mobile scenarios, which are inadequate for professional environments where documents must be carefully reviewed on full-size displays before signing.
3. **Lack of Real-Time Integrity Verification:** Few existing approaches address real-time document integrity checking during transfer. Mutual authentication with Secure Element credential storage to prevent man-in-the-middle substitution of document content during NFC transmission is absent from prior work.
4. **Network Dependence:** Most existing solutions require continuous internet connectivity. Insufficient research addresses maintaining full security guarantees in offline or intermittently connected environments, which are common in field healthcare, legal, and financial settings.

Table 3.1 presents a detailed comparison of existing digital signature solutions against the SLDS framework across eight key features.

Table 3.1: Comparison of Existing Digital Signature Solutions With The Proposed SLDS Framework.

Solution	Bidir. Transfer	Mutual Auth.	Digital Sig.	Cert. Mgmt.	Real-time Integrity	NFC Support	Cross- Platform	PC Signer
Ponnappalli & Saxena [16]	×	×	✓	✓	×	×	✓	✓
Weihua & Shuang [17]	×	×	✓	×	✓	×	×	✓
Yulianto et al. [18]	×	×	✓	✓	×	×	×	✓
Fun et al. [13]	×	×	✓	×	×	×	×	✓
Kishore et al. [19]	×	×	✓	×	×	×	×	✓
iDSign [6]	✓	✓	✓	×	×	✓	✓	×
SecureXfer [7]	✓	✓	✓	×	✓	✓	×	×
Proposed SLDS	✓	✓	✓	✓	✓	✓	✓	✓

3.3 System Architecture

SLDS is an NFC-enabled digital signature framework that eliminates the security vulnerabilities of remote document signing by creating a proximity-controlled signing ecosystem. The framework requires physical co-location of the communicating parties through NFC's 4–10 cm operating range, eliminating server-mediated vulnerabilities and network transmission risks. Figure 3.1 illustrates the complete SLDS system architecture.

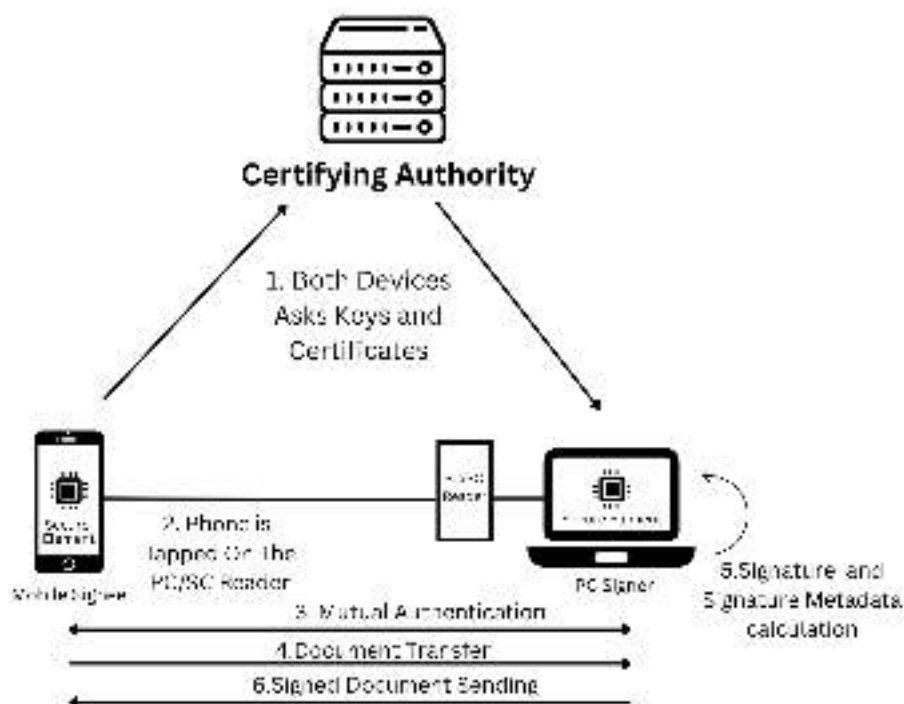


Figure 3.1: SLDS framework system architecture.

3.3.1 System Components

The SLDS framework comprises three principal components:

- **Android Device (Signee):** An NFC-enabled mobile device working in Host Card Emulation (HCE) mode (simulating a software contactless smart card). The Signee application holds the document to be signed, manages the whole NFC communication (mutual authentication and document transmission).

- **PC Workstation (Signer):** A desktop or laptop PC/laptop that connects to the USB PC/SC NFC reader. The Signer receives the document using NFC, presents it full-size for the Signee to review, applies an RSA-SHA256 digital signature and sends the signed document back to the Signee using NFC.
- **Certificate Authority (CA) Server:** Python based HTTP server that creates RSA-2048 key pair and X.509 certificates to register devices. The request received from the client with device identifier can be registration request. The CA creates a package with device certificate, private key, and CA certificate which is sent to the client for storing in to the client secure element.

Credentials for RSA-2048 are stored in Secure Elements on both of the parties involved, access is granted internally via SE specific APIs. All private key operations occur inside the SE so the application layer never gets to know the private key material.

3.4 Device Registration

Prior to each signing session, the PC Signer and the Android Signee need to be registered with the Certificate Authority (CA) server and obtain cryptographic identities from the CA. The procedure of registration is described below. Registration is a two-way operation between the two devices and the CA server. Both devices are required to first issue a registration request to the CA server containing their identifiers (device serial number or an e-mail address). After successfully verifying the identifiers of the requester, the CA server creates a 2048-bit RSA key pair for the requesting device and issues a X.509 certificate. This certificate binds the public key of the device to its verified identity, and the certificate is digitally signed by the private key of the CA. Finally, the CA server securely sends the entire credential package, which includes the device certificate, the device's private key, and the CA certificate, to the requesting device through a secure communication channel.

Upon receipt, each device stores the credentials in its Secure Element. The private key is stored within the SE in a form that allows cryptographic operations to be performed inside the SE without the key ever being exposed to the application processor. The

Table 3.2: Cryptographic Resource Sizes For The SLDS Framework.

Resource	Size (bytes)
RSA Public Key	256
RSA Private Key	256
X.509 Certificate	1024
Symmetric Session Key	128
Total per device	1664

device certificate and CA certificate are accessible to the application for presentation during mutual authentication. Table 3.2 presents the cryptographic resource sizes for the SLDS framework.

The total credential storage requirement of 1664 bytes per device represents less than 0.002% of typical mobile device memory, ensuring that the security infrastructure imposes negligible resource overhead.

3.5 Mutual Authentication Protocol

Table 3.3: Notations For The SLDS Mutual Authentication Protocol.

Notation	Meaning
P_{Signer}	Signer (PC Workstation / HCE Reader)
M_{Signee}	Signee (Android Device / HCE Card)
$P_{Signer_{cert}}$	X.509 Certificate of the Signer
$M_{Signee_{cert}}$	X.509 Certificate of the Signee
$P_{Signer_{pub}}$	RSA Public Key of the Signer
$P_{Signer_{pri}}$	RSA Private Key of the Signer
$M_{Signee_{pub}}$	RSA Public Key of the Signee
$M_{Signee_{pri}}$	RSA Private Key of the Signee
N_1, N_2	32-byte random nonces for mutual authentication
$SesK$	Symmetric Session Key for secure communication
$E_{A_{SK}}()$	Asymmetric Encryption using Key K
$E_{SK}()$	Symmetric Encryption using Key K
$D_{SK}()$	Symmetric Decryption using Key K
$Hash$	SHA-256 Hash of the file

The mutual authentication protocol of SLDS establishes verified identity and a shared session key between the PC Signer (P_{Signer}) and Android Signee (M_{Signee}) before any document is exchanged. The protocol employs a bidirectional challenge-response mechanism over the NFC APDU channel, as illustrated in Figure 3.2.

Table 3.3 defines the notation used throughout the protocol description. The mutual authentication protocol proceeds through the following steps:

1. **Authentication Initialization:** Upon NFC tap, P_{Signer} detects the Signee device and sends an authentication initialization command. M_{Signee} responds with an authentication readiness confirmation after loading its device certificates from the Secure Element.
2. **Certificate Exchange:** P_{Signer} sends a certificate request command to M_{Signee} , which responds with $M_{Signee_{cert}}$. P_{Signer} verifies the CA signature on the received certificate using the CA's public key. Upon

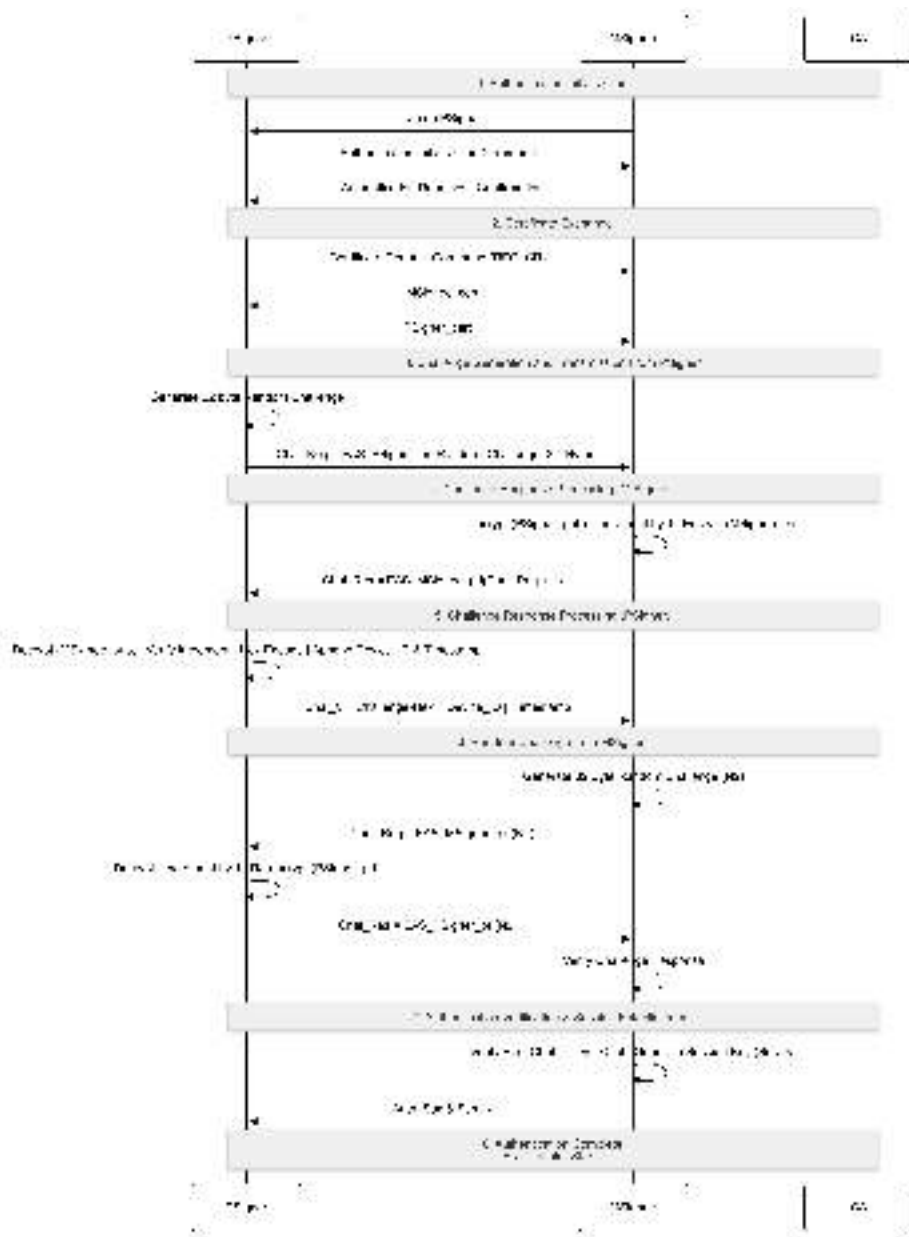


Figure 3.2: SLDS mutual authentication protocol sequence

successful verification, P_{Signer} transmits $P_{Signer_{cert}}$ to M_{Signee} , which similarly verifies the CA signature.

3. **Challenge Generation and Transmission from Signer:** P_{Signer} generates a 32-byte cryptographically random challenge N_1 , encrypts it using its own private key P_{Signer_pri} , and transmits it to M_{Signee} as

shown in Equation 3.1:

$$Chal_Req = EA_{P_{Signer_pri}}(N_1) \quad (3.1)$$

4. **Challenge Response from Signee:** M_{Signee} decrypts the received challenge using P_{Signer_pub} obtained from the verified certificate, increments the nonce by 1, encrypts the result using its own private key M_{Signee_pri} , and transmits the response as shown in Equation 3.2:

$$Chal_Res = EA_{M_{Signee_pri}}(N_1 + 1) \quad (3.2)$$

5. **Challenge Verification by Signer:** P_{Signer} decrypts the received response using M_{Signee_pub} , verifies the incremented nonce, concatenates the result with its device identifier and current timestamp, and transmits the verification result back to M_{Signee} as shown in Equation 3.3:

$$Chal_V = Challenge_Hex \parallel Device_ID \parallel Timestamp \quad (3.3)$$

6. **Reverse Challenge from Signee:** M_{Signee} performs a symmetric challenge-response in the reverse direction, generating its own 32-byte random nonce N_2 and following the same encryption, transmission, and verification procedure, proving possession of M_{Signee_pri} to P_{Signer} .
7. **Authentication Verification and Session Establishment:** Upon successful completion of both challenge-response exchanges, M_{Signee} confirms mutual authentication and generates a symmetric session key $SesK$ for all subsequent encrypted communication, as shown in Equation 3.4:

$$Auth_Suc = (Rec_Chal == Orig_Chal) \parallel Cert_V \quad (3.4)$$

The session key $SesK$ is transmitted from M_{Signee} to P_{Signer} encrypted under P_{Signer_pub} , ensuring that only P_{Signer} , with its SE-protected private key, can recover $SesK$. All subsequent NFC communication is encrypted using $SesK$.

3.6 Secure File Transfer

Following successful mutual authentication, the secure file transfer process begins. The Signee prepares the document for transmission through a se-

ries of steps that ensure confidentiality and integrity throughout the NFC transfer.

1. **File Hash Calculation:** M_{Signee} computes the SHA-256 hash of the original document to establish a reference integrity value for subsequent verification, as shown in Equation 3.5:

$$Original_File_Hash = Hash(Original_File) \quad (3.5)$$

2. **Metadata Transfer:** M_{Signee} creates a metadata package containing the file format, file type, file size, and computed hash, and transmits this to P_{Signer} for preprocessing and validation, as shown in Equation 3.6:

$$Metadata = File_Format \parallel File_Type \parallel File_Size \parallel File_Hash \quad (3.6)$$

3. **File Encryption:** M_{Signee} encrypts the original file using the established session key $SesK$ to ensure confidentiality during NFC transmission, as shown in Equation 3.7:

$$Encrypted_File = ES_{SesK}(Original_File) \quad (3.7)$$

4. **Chunked Transmission:** M_{Signee} divides the encrypted file into sequential APDU-sized chunks and transmits them over the NFC channel, with the interface displaying real-time progress to the user, as shown in Equation 3.8:

$$Transmission = Chunk_1 \parallel Chunk_2 \parallel \dots \parallel Chunk_n \quad (3.8)$$

5. **File Decryption:** Upon receiving all chunks, P_{Signer} reconstructs the complete encrypted file and decrypts it using $SesK$, as shown in Equation 3.9:

$$Decrypted_File = DS_{SesK}(Received_Encrypted_File) \quad (3.9)$$

6. **Integrity Verification:** P_{Signer} computes the SHA-256 hash of the received and decrypted file and compares it against the originally transmitted hash value, confirming successful and unmodified transmission,

as shown in Equation 3.10:

$$Verification_Success = Hash(Decrypted_File) == Original_File_Hash \quad (3.10)$$

If integrity verification fails, the transfer is aborted and the Signee is notified. Only upon successful hash verification does the Signer proceed to the digital signing phase.

3.7 Digital Signing and Verification

After securely receiving and verifying the document, P_{Signer} performs the digital signing process and returns the signed document to M_{Signee} .

1. **Digital Signature Generation:** P_{Signer} computes the SHA-256 hash of the decrypted document and encrypts this hash using its RSA private key P_{Signer_pri} to create the digital signature, ensuring authenticity and non-repudiation, as shown in Equation 3.11:

$$Digital_Signature = EA_{P_{Signer_pri}}(Hash(Decrypted_File)) \quad (3.11)$$

2. **Signature Metadata Creation:** P_{Signer} creates comprehensive signature metadata containing the digital signature, current timestamp, file size, and Signer information, and appends this as structured content to the original document, as shown in Equation 3.12:

$$Signed_Document = Original_File \parallel Digital_Signature \parallel Timestamp \parallel Signer_Info \quad (3.12)$$

3. **Document Re-encryption and Transmission:** P_{Signer} encrypts the complete signed document using $SesK$ and transmits it back to M_{Signee} over the NFC channel, as shown in Equation 3.13:

$$Encrypted_Signed_Doc = ES_{SesK}(Signed_Document) \quad (3.13)$$

4. **Signature Verification:** Upon receiving the signed document, M_{Signee} decrypts it using $SesK$, then verifies the digital signature by decrypting it using P_{Signer_pub} from the verified certificate and comparing the result to an independently computed hash of the original document, as

shown in Equation 3.14:

$$Signature_Verified = DA_{P_{Signer_pub}}(Digital_Signature) == Hash(Original_File) \quad (3.14)$$

If signature verification succeeds, the signed document is stored locally on the Signee device. The embedded digital signature provides long-term verifiability: any party in possession of P_{Signer_pub} can independently verify the signature at any future time without requiring the CA server or any network connectivity. Figure 3.3 illustrates the SLDS framework flow diagram.

3.8 Novelty of SLDS

SLDS introduces three novel contributions that distinguish it from all existing digital signature frameworks:

1. **Proximity-Based Security with Locality Proof:** SLDS physically compels proximity through NFC's 4–10 cm operating range, negating the remote attack surface and providing an implicit guarantee of physical co-location that is critical for healthcare prescription signing, legal contract execution, and financial authorization workflows. The locality-of-proof property ensures that document signing is contingent upon the physical co-location of the communicating parties, rendering man-in-the-middle attacks infeasible within the NFC transmission range.
2. **Mutual Authentication with Certificate Authority:** SLDS implements full bidirectional mutual authentication in which both devices authenticate themselves through CA-issued X.509 certificates stored in tamper-resistant Secure Elements. The challenge-response protocol makes sure only devices possessing valid CA-issued credentials, along with the protected private key in the SE, can access the signing process. This eliminates the risk of impersonation by unauthorized devices and provides a verifiable chain of trust back to the CA.
3. **PC-Based Document Review:** The large-screen workstation interface of the PC Signer allows thorough examination of documents before signing, providing a significant usability and informed consent advantage over mobile-only solutions. Professional document signing workflows require that the signer read and understand the complete document before committing a signature, a requirement that mobile screen sizes cannot adequately support for multi-page documents.

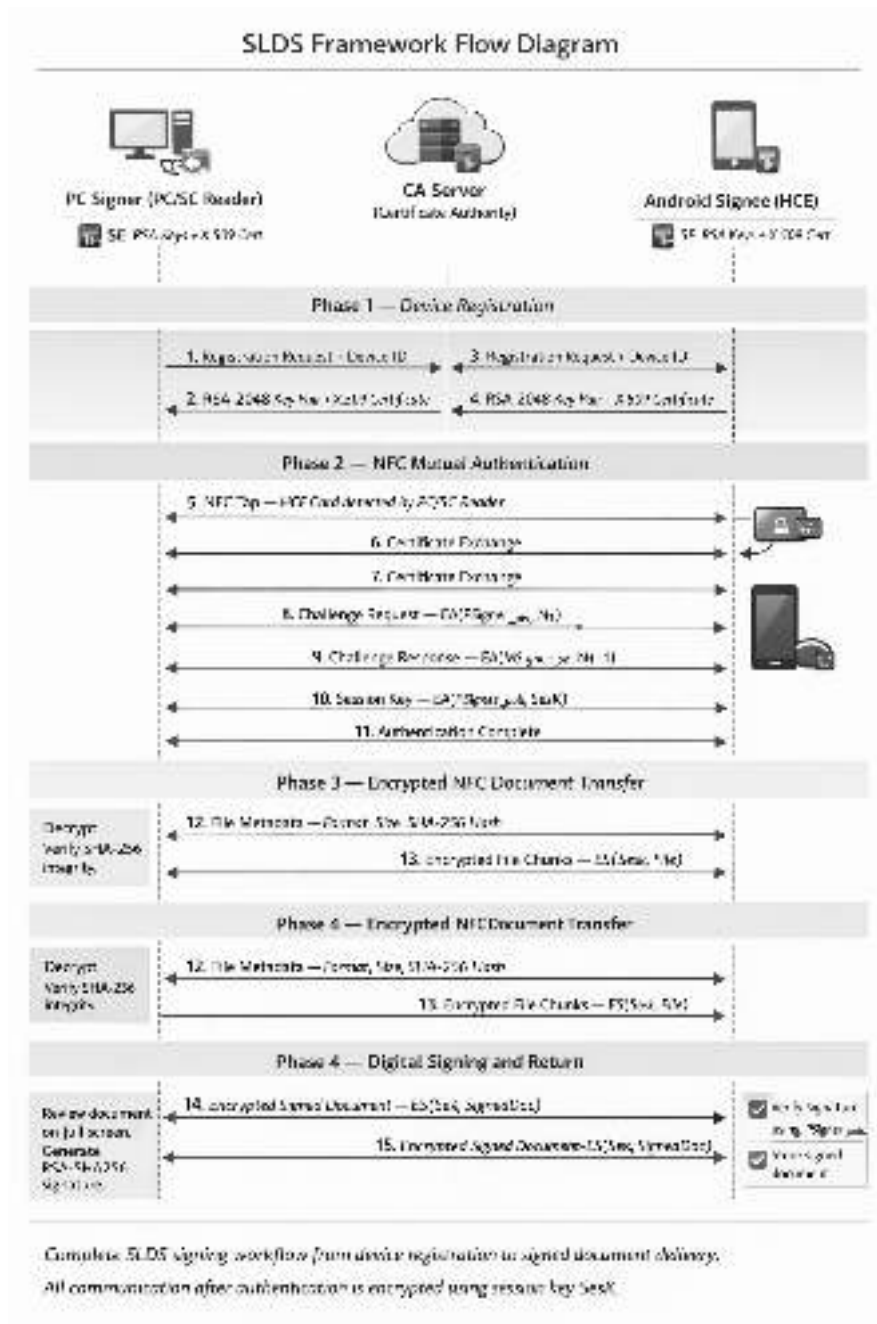


Figure 3.3: SLDS framework flow diagram.

3.9 Security Analysis

This section evaluates the security mechanisms of SLDS against the principal threat vectors relevant to proximity-based digital signing systems.

1. Man-in-the-Middle Attack Resistance: SLDS mitigates man-in-the-middle attacks through two complementary mechanisms. At the physical layer, NFC’s 4–10 cm operating range requires any eavesdropping adversary to be physically present within centimetres of the communicating devices, making covert interception practically infeasible in normal operating environments. At the cryptographic layer, the bidirectional challenge-response protocol ensures that any modification of authentication messages is detected: each challenge is encrypted under the generating party’s private key and verified by the peer using the CA-issued certificate, so a man-in-the-middle cannot substitute or replay authentication messages without possession of an SE-protected private key.

2. Replay Attack Resistance: The mutual authentication protocol employs 32-byte cryptographically random nonces N_1 and N_2 that are unique to each authentication session. The challenge-response mechanism requires each party to demonstrate knowledge of the received nonce by incrementing it and returning the incremented value encrypted under its own private key. Timestamp verification in the challenge verification step further ensures that captured authentication messages cannot be replayed in subsequent sessions, as the timestamp comparison will fail for any message captured in a prior session.

3. Secure Element Storage: RSA-2048 private keys are stored in tamper-resistant Secure Elements that execute all cryptographic operations within an isolated hardware environment. Even if the host device is compromised at the operating system or application layer, the private key material remains inaccessible to any unauthorized entity. This hardware isolation ensures that credential integrity is maintained regardless of the security state of the host processor.

4. Session Key Protection: Following mutual authentication, a symmetric session key $SesK$ takes over for encrypting all subsequent file transfers. The session key is transmitted from M_{Signee} to P_{Signer} encrypted under $P_{Signer_{pub}}$, ensuring that only P_{Signer} , with its SE-protected private key, can recover $SesK$. A fresh session key is generated for each signing session, ensuring that compromise of one session key does not affect the security of past or future sessions.

5. Relay Attack Resistance: Relay attacks convey NFC messages between distant devices using a high-bandwidth channel, creating a false impression of proximity [8]. SLDS defeats relay attacks through SE-based credential storage: since SE-protected private keys are never exposed, relay proxies cannot generate valid cryptographic responses to authentication challenges. Furthermore, fresh SE-bound nonces and encrypted tokens bind each authentication session to specific registered Secure Elements, preventing

external proxies from computing the necessary cryptographic responses [12].

6. Non-Repudiation: The RSA-SHA256 digital signature embedded in the signed document provides long-term non-repudiation guarantees. The signature can be verified by any party holding $P_{Signer_{pub}}$ at any future time, independently of the CA server or any network connectivity. The embedded timestamp and Signer information further establish the time and identity of the signing event.

3.10 Formal Verification Using Scyther

In this thesis, we will give a formal proof of the correctness of the SLDS mutual authentication protocol, especially considering its security against symbolic protocol attacks. For the purpose of verification, we use the automatic verification tool Scyther [14].

Scyther gives a full symbolic analysis assuming that cryptography is computationally secure and the adversary control the entire network in the context of Dolev-Yao model.

3.10.1 SPDL Model

The SLDS mutual authentication protocol is modelled in the Security Protocol Description Language (SPDL). The model specifies three roles: Signer (P_{Signer}), Signee (M_{Signee}), and Certificate Authority (CA), each possessing RSA-2048 key pairs and CA-signed X.509 certificates. The model captures the complete eight-message authentication sequence: NFC initiation, certificate exchange, Signer challenge generation and transmission, Signee challenge response, Signee reverse challenge, Signer reverse response, authentication verification, and session key distribution.

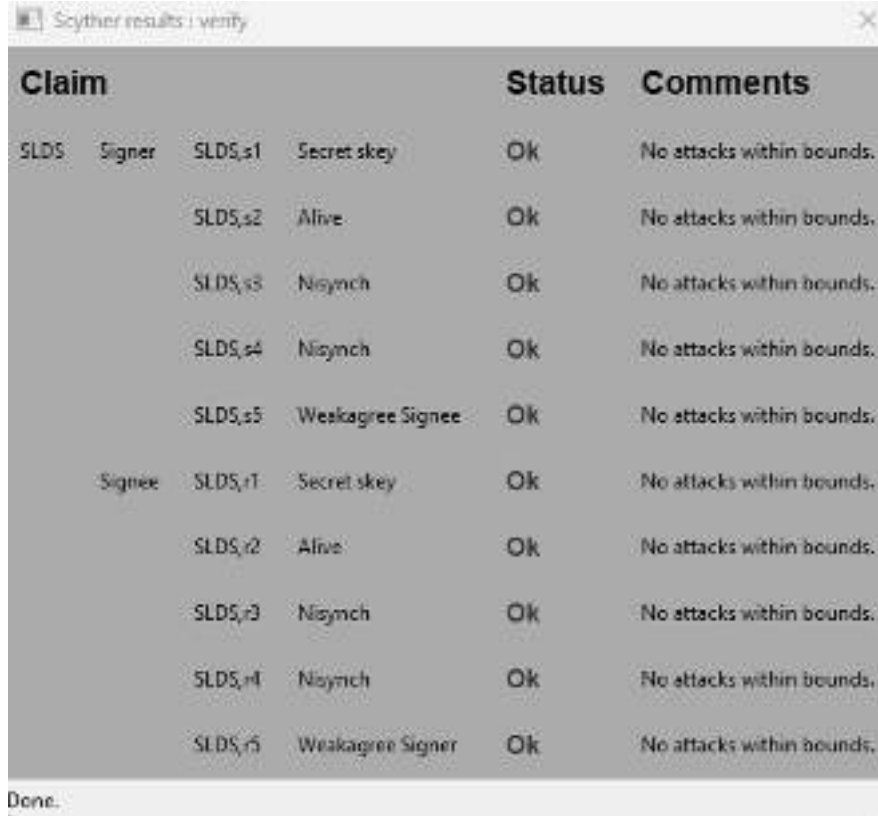
Both the Signer and Signee roles claim the following security properties in the SPDL model:

- **Secret key ($SesK$):** The session key must remain unknown to the adversary.
- **Aliveness:** The peer role must have been executed at least once in every complete run.
- **Non-injective synchronization (Nisynch):** Both roles must agree on all exchanged message terms in the same session.
- **Nonce commitment:** Generated nonces must be committed to the specific session.

- **Weak agreement:** Each role must agree on the identity of its peer.

3.10.2 Verification Results

Figure 3.4 presents the Scyther verification results for the SLDS protocol.



Claim				Status	Comments
SLDS	Signer	SLDS,s1	Secret key	Ok	No attacks within bounds.
		SLDS,s2	Alive	Ok	No attacks within bounds.
		SLDS,s3	Nisynch	Ok	No attacks within bounds.
		SLDS,s4	Nisynch	Ok	No attacks within bounds.
		SLDS,s5	Weakagree Signee	Ok	No attacks within bounds.
	Signee	SLDS,r1	Secret key	Ok	No attacks within bounds.
		SLDS,r2	Alive	Ok	No attacks within bounds.
		SLDS,r3	Nisynch	Ok	No attacks within bounds.
		SLDS,r4	Nisynch	Ok	No attacks within bounds.
		SLDS,r5	Weakagree Signer	Ok	No attacks within bounds.

Done.

Figure 3.4: Scyther verification results for the SLDS mutual authentication protocol.

Scyther confirms no attacks across unbounded concurrent sessions, verifying all ten security claims for both roles. Specifically:

- **Session key secrecy** is preserved because the session key is encrypted with $P_{Signer_{pub}}$ and decryptable only using the SE-protected $P_{Signer_{pri}}$, which is never exposed to the network.
- **Aliveness:** The peer role must have executed at least once in all successful runs.
- **Nisynch:** The roles must concur on all transmitted message terms during the same session.

- Nonce commitment: Produced nonces must commit to the correct session.
- Weak agreement: Roles must agree on who their peer is.

Coupling this formally verified authentication protocol with hardware-based Secure Elements, CA-issued certificate validation and NFC proximity requirements allows the SLDS proximity based digital signing workflow to have a strong security backbone.

3.11 Performance Analysis

Table 3.4 outlines the timing for the end to end SLDS digital signature process for a 500KB PDF file showing processing cost and transfer time.

Table 3.4: SLDS End-to-End Timing Analysis For A 500 KB PDF Document.

Process	Time (seconds)
Mutual Authentication	5.0
File Metadata Transfer	4.0
File Sending (Signee $\rightarrow$ Signer, NFC)	120.0
Digital Signing and Embedding	1.5
File Sending (Signer $\rightarrow$ Signee, NFC)	120.0
Signature Verification	3.0
Total	253.5

As seen from the results, SLDS provides a strong security assurance without significant cryptographic overhead. The 5 second time span is considered to cover the certificate exchange, both challenge response authentication operations and the RSA based cryptographic check. The time required to sign, where digital signature needs to be generated by using RSA-SHA256 signature creation followed by signature being appended into PDF document takes 1.5 sec which is 0.6% of overall transaction. Then signature verification needs to be carried out where hash has been recalculated and certificate chain verification carried out which takes 3 sec time span.

Total transaction time is 253.5 sec where time taken to transfer the file bidirectionally using NFC is 240 sec (94.7% of total transaction time). All the

security operations together takes only 13.5 sec time, this is irrespective of the file size. Therefore as the file size increase it provides more security ratio to the transfer, hence increasing the file size leads to the SLDS-E (Chapter 4).

3.12 Use Case: Healthcare Insurance Prescription Signing

Healthcare insurance fraud currently generates billions of dollars of damages each year via medical record manipulation, forged prescriptions, or treatment record changes [20]. Digital signatures provide a method of cryptographic authentication and involve tamper-evident seals to prevent the tampering of documents post signing. This attribute provides unique benefits for proximity-based signing schemes such as those within the healthcare sector wherein proof of physical proximity for the physician is a mandatory requirement.

Extending the NFC-based health folder S-MAPLE system [21], SLDS implements the following process. The patients would carry around mobile devices that act as contact-less health wallets containing their medical records, prescription records, treatment history and etc. When a physician must sign a prescription or treatment authorization, the patient presents their mobile device for an NFC tap at the physician's workstation PC/SC reader. The complete SLDS workflow executes: mutual authentication establishes cryptographic proof that the physician's registered workstation is physically co-located with the patient's device; the prescription document is transferred over NFC to the physician's workstation for full-screen review; the physician reviews the complete document and applies an RSA-SHA256 digital signature; and the signed prescription is returned to the patient's mobile health wallet.

Insurance providers verify submitted claims by checking the RSA-SHA256 digital signature against the physician's X.509 certificate. This confirms the authenticity of the signature and the identity of the signing physician. The locality-of-proof guarantee of SLDS ensures that the signature could only have come from a properly registered physician device near the patient's device. This helps eliminate the most frequent ways prescriptions are forged from a distance and without authentic signatures.

Chapter 4

Proposed SLDS-E Application

4.1 Introduction

The SLDS framework fulfills the basic demands: proximity-enforced digital signing with PC based document review, and formally verified mutual authentication. The actual measured performance indicates one operational bottleneck of the proposed SLDS framework: data throughput is restricted to 40–50 kbps due to using NFC for transmitting large documents in bulk; which results in 253.5 seconds for one signing transaction (e.g., 500 KBs) and more than 60 minutes for 5 MBs of document to be signed. The bottleneck is not due to defects of the SLDS protocol design. It comes from limitations of physical layer NFC radio channel and cannot be addressed by software.

This chapter presents Secure Location DigiSign Enhanced (SLDS-E), a hybrid proximity digital signing framework that resolves the NFC bandwidth bottleneck of SLDS by decoupling mutual authentication from bulk document transfer. SLDS-E retains NFC exclusively for proximity-enforced mutual authentication and Wi-Fi Direct credential exchange, preserving the complete security foundation of SLDS without modification, while delegating all bulk document transfer to a high-speed TCP channel over Wi-Fi Direct secured with AES-256-GCM authenticated encryption using the NFC-negotiated session key. Wi-Fi Direct network credentials are delivered exclusively over the authenticated NFC channel, ensuring that physical co-location at the NFC tap remains a mandatory prerequisite for initiating any document transfer, regardless of the transport channel used.

Experimental evaluation demonstrates throughput improvements of 40 times to 720 times over pure NFC transfer across document sizes from 200 KB to 5 MB, reducing a 500 KB signing session from 253.5 seconds to 10 seconds — a 96% reduction in end-to-end latency — while preserving all security

guarantees of the original SLDS framework including mutual authentication, proximity enforcement, and cryptographic non-repudiation.

The chapter is organized as follows. Section 4.2 reviews related work and identifies the research gaps addressed by SLDS-E beyond those addressed by SLDS. Section 4.3 presents the system architecture. Sections 4.4 through 4.7 describe the four operational phases of SLDS-E in detail. Section 4.8 discusses the novelty contributions. Section 4.9 presents the security continuity analysis confirming that no new attack surfaces are introduced. Section 4.10 presents the performance evaluation. Section 4.12 presents the healthcare use case demonstrating the practical impact of the throughput improvement.

4.2 Related Work and Research Gaps

4.2.1 Limitations of Existing Proximity Signing Frameworks

The analysis of existing proximity signing frameworks presented in Chapter 3 identified four research gaps addressed by SLDS. SLDS-E addresses three additional gaps that emerge specifically from the throughput limitations of pure NFC-based document transfer.

The foundational SLDS framework [22] addressed the locality proof, PC-based signing, certificate management, and real-time integrity verification gaps identified in Chapter 3. However, SLDS retains NFC as the sole file transfer channel, causing transfer times that grow linearly with document size and reach operationally infeasible values for documents larger than a few hundred kilobytes. This limitation motivates the hybrid architecture of SLDS-E.

SecureXfer [7] demonstrated that decoupling authentication from file transfer improves performance, achieving a total signing time of 12.5 seconds for a 1 MB document by using NFC for mutual RSA-based authentication and Bluetooth Low Energy for file transfer. However, SecureXfer is constrained to mobile-to-mobile communication and does not support PC-based signing. Furthermore, SecureXfer does not reuse the NFC authentication session key for encrypting the Bluetooth transfer channel, creating a cryptographic discontinuity between the authentication and transfer phases.

iDSign [6] employs Identity-Based Cryptography with a modified iTLS protocol, achieving a 97% reduction in cryptographic overhead compared to certificate-based approaches, but is similarly constrained to mobile-to-mobile scenarios and uses NFC as the sole transfer channel.

4.2.2 Research Gaps Addressed by SLDS-E

SLDS-E addresses the following gaps that remain unresolved in existing literature after the contributions of SLDS:

1. **NFC Bandwidth Bottleneck:** No existing proximity signing framework resolves the fundamental throughput limitation of NFC as a file transfer channel. Transfer times exceeding 60 minutes for a 5 MB document make NFC-only systems operationally infeasible for enterprise deployment with real-world document sizes including medical imaging records and multi-page legal contracts.
2. **Lack of High-Speed Proximity Transfer:** SecureXfer demonstrated the value of decoupling authentication from transfer, but Bluetooth throughput remains limited compared to Wi-Fi Direct and the framework is constrained to mobile-to-mobile configurations. No existing framework combines NFC proximity authentication with Wi-Fi Direct bulk transfer in a cross-platform PC-to-mobile configuration.
3. **Session Key Discontinuity:** Existing hybrid frameworks do not reuse the authentication session key for encrypting the file transfer channel, creating a security gap between the authenticated session and the transfer session. SLDS-E eliminates this discontinuity by reusing the NFC-negotiated session key directly for AES-256-GCM encryption of all Wi-Fi Direct TCP packets.
4. **Cross-Platform High-Throughput PC-to-Mobile Signing:** No prior framework simultaneously supports PC-based signing with large-screen document review, NFC-enforced proximity authentication, end-to-end session key continuity, and high-speed wireless file transfer across heterogeneous operating systems — Windows PC and Android — without platform-specific drivers.

4.3 System Architecture

SLDS-E extends the SLDS architecture by introducing a Wi-Fi Direct communication layer alongside the existing NFC layer. The framework operates in two distinct sequential phases: the NFC Authentication Phase, which executes the complete SLDS mutual authentication protocol without modification, and the Wi-Fi Direct Transfer Phase, which replaces the NFC file transfer channel of SLDS with a high-throughput AES-256-GCM secured

TCP connection. Figure 4.1 illustrates the complete SLDS-E system architecture.

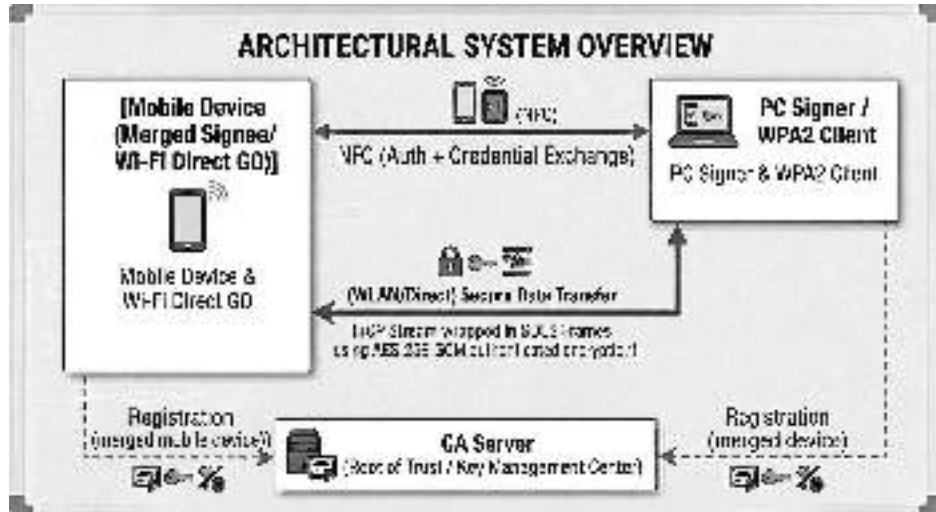


Figure 4.1: SLDS-E system architecture showing the two-channel hybrid communication model.

Table 4.1 defines the notation used throughout the SLDS-E protocol description, extending the SLDS notation of Table 3.3 with Wi-Fi Direct and AES-256-GCM specific terms.

Table 4.1: Additional Notations For The SLDS-E Framework Beyond Those defined For SLDS.

Notation	Meaning
$SesK$	AES-256 Session Key (32 bytes), negotiated during NFC auth
IV	Initialization Vector (12 bytes), unique per TCP packet
Tag	AES-GCM Authentication Tag (16 bytes)
$SSID$	Wi-Fi Direct Network Name
PSK	Wi-Fi Direct WPA2 Passphrase
$Port$	TCP Server Port (8765)
C_{info}	Wi-Fi Direct Credential Bundle: $SSID\ PSK\ Port$
$E_K^{GCM}()$	AES-256-GCM Encryption using Key K
$D_K^{GCM}()$	AES-256-GCM Decryption using Key K
F	Original Document File
F_{signed}	Digitally Signed Document
$MD5$	MD5 Checksum for transfer integrity verification

4.3.1 System Components

The SLDS-E system comprises three components, extending the SLDS components with Wi-Fi Direct capability:

- **Android Device (Signee):** A mobile device with NFC capability running the SLDS-E Signee application in HCE mode. In addition to the SLDS HCE service, the Signee application activates a Wi-Fi Direct Group Owner network and operates a TCP server on port 8765 prior to the NFC tap, ready to accept incoming document transfer connections following authentication.
- **PC Workstation (Signer):** A desktop or laptop computer with a PC/SC NFC reader for mutual authentication. Following authentication, the PC joins the Android-hosted Wi-Fi Direct network as a standard WPA2 IEEE 802.11 infrastructure client using the Windows Native WiFi API (WlanConnect) and establishes a TCP connection to the Signee's server for AES-256-GCM secured document transfer.
- **Certificate Authority (CA) Server:** The Python-based CA server is identical to the SLDS CA architecture, generating RSA-2048 key

pairs and X.509 certificates for registered devices. No modifications to the CA are required for SLDS-E.

4.4 Phase 1: NFC Mutual Authentication

SLDS-E retains the complete mutual authentication protocol of SLDS without any modification. Both devices register with the CA server to obtain RSA-2048 key pairs and X.509 certificates, stored in tamper-resistant Secure Elements. Upon NFC tap, both devices execute the complete bidirectional challenge-response protocol described in Section 3.5, verifying CA-issued certificates and proving private key possession through nonce exchange.

Apart from SLDS authentication phase, SLDS-E only adds session key generation step. A 32-byte AES-256 session key $SesK$ is generated by Signee and encrypted by Signer's RSA public key with OAEP-SHA256 padding then sending as the last APDU in the authentication sequence as following:

$$SesK_{enc} = EA_{P_{Signer_{pub}}}(SesK) \quad (4.1)$$

The Signer uses its SE-protected private key to decrypt $SesK_{enc}$ to get $SesK$. Both devices share only $SesK$ at this moment and will use this key for encryption of future Wi-Fi Direct TCP packets (all messages sent between the devices after this point) in AES-256-GCM which links NFC to Wi-Fi Direct.

4.5 Phase 2: Wi-Fi Direct Credential Exchange

Once mutual authentication is complete, the Signee sends the Wi-Fi Direct network details over the secured NFC link to the Signer, using a special APDU command with instruction byte $INS = 0xE0$. The credential bundle C_{info} is represented by:

$$C_{info} = SSID \parallel PSK \parallel Port \quad (4.2)$$

The Android Signee has setup the Wi-Fi Direct Group Owner network and TCP server on $Port = 8765$ before the NFC tap occurs, such that when the credential arrives over the Wi-Fi Direct connection after the NFC tap the network is already waiting. Delivery of C_{info} only over the authenticated NFC channel is the mechanism that ties the Wi-Fi Direct transfer session with the NFC tap. As NFC operates within 4–10 cm, only a device that is physically co-located with the Android Signee and has successfully completed

the full RSA-based challenge-response authentication receives the credentials necessary to join the Wi-Fi Direct network. A remote adversary has no path to obtain C_{info} without physically tapping the Signee device and passing the complete mutual authentication protocol.

4.6 Phase 3: Wi-Fi Direct Network Association

Upon receiving C_{info} , the PC Signer constructs a WPA2-Personal WLAN profile using the received $SSID$ and PSK and joins the Android-hosted Wi-Fi Direct group as a standard IEEE 802.11 infrastructure client using the Windows Native WiFi API. The WLAN profile follows WPA2-PSK authentication with AES cipher mode:

$$Profile = \{SSID, WPA2PSK, AES, PSK\} \quad (4.3)$$

This approach exploits the Wi-Fi Direct Group Owner's compatibility with legacy IEEE 802.11 infrastructure clients [23]. The PC Signer joins the group without requiring Wi-Fi Direct peer-to-peer negotiation or any platform-specific drivers, using only the standard Windows WLAN API available on all modern Windows installations. Following successful network association, the PC Signer establishes a TCP connection to the Signee's server at *Port* 8765, completing the channel setup for AES-256-GCM secured document transfer.

4.7 Phase 4: AES-256-GCM Secured Document Transfer

All document data exchanged over the Wi-Fi Direct TCP channel is encrypted using AES-256-GCM with the session key $SesK$ established during NFC mutual authentication. Each encrypted TCP packet is structured as:

$$Packet = IV \parallel E_{SesK}^{GCM}(Data) \parallel Tag \quad (4.4)$$

where IV is a 12-byte cryptographically random Initialization Vector unique to each packet, $E_{SesK}^{GCM}(Data)$ is the AES-256-GCM ciphertext, and Tag is the 128-bit Galois/Counter Mode authentication tag providing integrity and authenticity verification per packet. The transfer protocol proceeds through the following steps:

1. **File Request:** The PC Signer sends a `REQUEST_FILE` opcode (0x01) to the Signee TCP server to initiate the transfer session.
2. **File Payload Transmission:** The Signee transmits a structured payload containing the file size, filename, MD5 checksum, and AES-256-GCM encrypted file data:

$$Payload = |F| \parallel Name \parallel MD5(F) \parallel E_{SesK}^{GCM}(F) \quad (4.5)$$

3. **Integrity Verification and Digital Signing:** The PC Signer decrypts the received file using $SesK$ and verifies the MD5 checksum for transfer integrity. Upon successful verification, the Signer computes the RSA-SHA256 digital signature and embeds it into the document to produce F_{signed} , following the signing procedure described in Section 3.7.
4. **Signed Document Return:** The Signer transmits a `SIGNED_PAYLOAD` opcode (0x02) followed by the signed document encrypted under the same session key:
$$Return = E_{SesK}^{GCM}(F_{signed}) \quad (4.6)$$
5. **Verification and Acknowledgement:** The Signee decrypts F_{signed} using $SesK$, verifies the RSA-SHA256 digital signature using the Signer's X.509 certificate public key following Equation 3.14, and transmits an `ACK` opcode (0x06) to complete the transfer session upon successful verification.

Figure 4.2 illustrates the complete four-phase SLDS-E protocol sequence from NFC tap to signed document delivery.

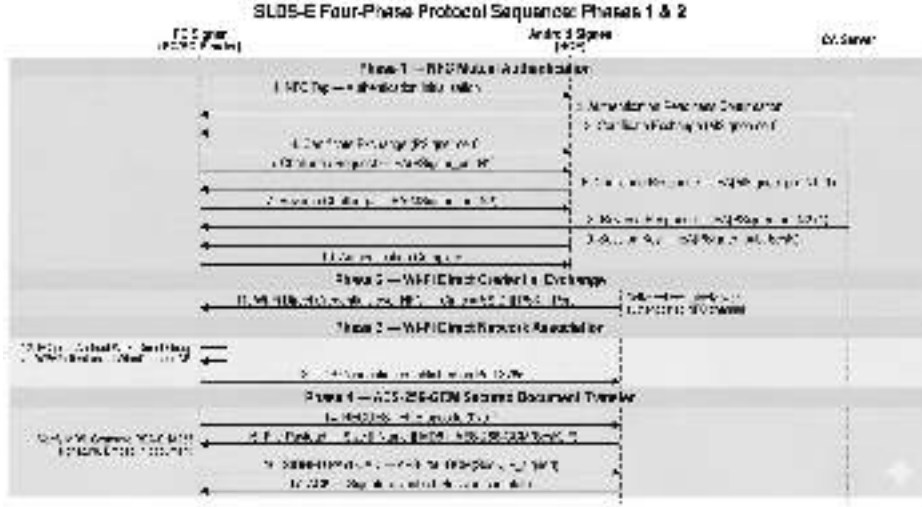


Figure 4.2: Complete SLDS-E four-phase protocol sequence.

4.8 Novelty of SLDS-E

SLDS-E introduces three novel contributions beyond the SLDS framework:

1. **Proximity-Bound High-Speed Transfer Channel:** SLDS-E is the first proximity signing framework to bind a Wi-Fi Direct transfer channel to an NFC authentication session through in-band credential delivery. By transmitting C_{info} exclusively over the authenticated NFC APDU channel, SLDS-E ensures that Wi-Fi Direct network access is contingent on physical co-location and successful mutual authentication. A remote adversary who intercepts Wi-Fi Direct transmissions cannot recover plaintext document data without $SesK$, and cannot obtain $SesK$ or C_{info} without completing the full NFC-based mutual authentication at the physical tap location.
2. **Session Key Continuity Across Channels:** SLDS-E reuses the AES-256-GCM session key $SesK$ negotiated during NFC mutual authentication directly for encrypting all Wi-Fi Direct TCP packets. This design eliminates any cryptographic discontinuity between the authentication and transfer phases: the same key that authenticated the session protects all document data, ensuring that the security established during NFC authentication extends seamlessly to the Wi-Fi Direct transfer channel without any additional key negotiation or trust establishment step.

3. **Cross-Platform Legacy Client Integration:** Due to the IEEE 802.11 infrastructure support for the Group Owner in Wi-Fi Direct, SLDS-E uses the Group Owner’s capability to allow the connection of a Windows computer in the Wi-Fi Direct network managed by the Android device using the standard native Windows WiFi API interface. The implementation of this approach avoids the requirement of Wi-Fi Direct drivers for the Signer platform. Thus, SLDS-E can be used on any modern Windows computer with the PC/SC NFC reader card, which is available in regular enterprise workstations.

4.9 Security Analysis

This section demonstrates that the Wi-Fi Direct transfer channel of SLDS-E preserves all security properties of the foundational SLDS framework without introducing any new attack surfaces.

1. Preservation of Proximity Enforcement: SLDS-E preserves the locality-of-proof guarantee of SLDS through in-band delivery of Wi-Fi Direct credentials over the authenticated NFC channel. Since C_{info} — comprising the Wi-Fi Direct *SSID*, *PSK*, and *Port* — is transmitted exclusively via the NFC APDU channel after successful mutual authentication, a remote adversary cannot obtain these credentials without physically tapping the Android Signee and passing the full RSA-based challenge-response protocol. Physical co-location within 4–10 cm therefore remains a mandatory prerequisite for initiating any document transfer, regardless of whether that transfer subsequently occurs over NFC or Wi-Fi Direct.

2. Inheritance of SLDS Mutual Authentication: SLDS-E executes the complete SLDS mutual authentication protocol without any modification. The formal Scyther verification results presented in Section 3.10 therefore apply directly to SLDS-E: all ten security claims including session key secrecy, aliveness, non-injective synchronization, nonce commitment, and weak agreement hold for the SLDS-E authentication phase under the Dolev-Yao adversary model.

3. Wi-Fi Direct Channel Security: The Wi-Fi Direct group operates under WPA2-Personal security, providing link-layer confidentiality for all wireless transmissions. SLDS-E independently applies AES-256-GCM encryption to every TCP packet using $SesK$, providing a second independent layer of cryptographic protection at the application layer. Even an adversary who compromises the WPA2 link-layer encryption — for example, by recovering the WPA2 passphrase from C_{info} — cannot recover plaintext document data without also possessing $SesK$, which is never transmitted over any wire-

less channel. The 128-bit GCM authentication tag per packet detects any tampering, truncation, or reordering of transferred data with a probability of $1 - 2^{-128}$ of an undetected attack, and the MD5 checksum provides an additional application-layer integrity check at the file level.

4. Session Key Protection: The session key $SesK$ is generated by the Android Signee, encrypted using the PC Signer’s RSA public key with OAEP-SHA256 padding, and transmitted over the NFC channel during the final authentication step. $SesK$ never appears in plaintext over any wireless channel. Each packet carries a unique 12-byte IV generated by a cryptographically secure random number generator, preventing IV reuse within a session. A fresh $SesK$ is generated for each NFC authentication session, ensuring that compromise of one session’s key does not affect any other session.

5. Relay Attack Resistance: SLDS-E inherits relay attack resistance from SLDS through SE-based mutual authentication. Since all cryptographic operations execute within tamper-resistant Secure Elements and private keys are never exposed, relay proxies cannot generate valid authentication challenge responses [12]. Furthermore, a successful relay attack on the NFC authentication channel alone is insufficient to compromise SLDS-E: the Wi-Fi Direct credentials C_{info} are delivered post-authentication and bind the transfer session to the physical location of the NFC tap. Only the device physically present at the tap location receives C_{info} and can initiate the TCP connection to the Signee’s server.

6. No New Attack Surfaces: The transition from NFC to Wi-Fi Direct for bulk transfer introduces no new attack surfaces relative to SLDS. The Wi-Fi Direct network is inaccessible to any device that did not physically complete the NFC authentication tap, as C_{info} is the only path to obtaining the network credentials. Application-layer AES-256-GCM encryption ensures that link-layer compromise of Wi-Fi Direct does not compromise document confidentiality. The integrity of every transferred packet is verified through the GCM authentication tag before any decrypted data is processed.

4.10 Performance Evaluation

4.10.1 Resource Efficiency

SLDS-E inherits the cryptographic resource profile of SLDS without modification, as the mutual authentication protocol and RSA key sizes are unchanged. Table 4.2 presents the complete cryptographic storage requirements per device for SLDS-E, including the additional AES-256-GCM parameters introduced for the Wi-Fi Direct transfer channel.

Table 4.2: Cryptographic Resource Sizes For SLDS-E Per Device.

Resource	Size (bytes)
RSA Public Key	256
RSA Private Key	256
X.509 Certificate	1024
AES-256-GCM Session Key	32
AES-GCM IV per Packet	12
AES-GCM Authentication Tag per Packet	16
Total Cryptographic Overhead	1596

The total cryptographic storage requirement of 1596 bytes per device is marginally lower than the 1664 bytes required by SLDS, as the AES-256-GCM session key replaces the larger symmetric session key of SLDS. The complete cryptographic framework requires less than 0.002% of typical mobile device memory, with fixed resource requirements that remain constant regardless of file size or transfer frequency.

4.10.2 Transfer Throughput Comparison

Table 4.3 presents measured transfer times for SLDS and SLDS-E across four representative document sizes, demonstrating the throughput improvement introduced by replacing the NFC transfer channel with Wi-Fi Direct.

Table 4.3: Transfer Time Comparison Between SLDS (NFC) And SLDS-E (Wi-Fi Direct) Across Document Sizes.

File Size	SLDS — NFC	SLDS-E — Wi-Fi Direct	Speedup
200 KB	~2 min	~3 sec	40×
500 KB	~6 min	~3 sec	120×
1 MB	~12 min	~5 sec	144×
5 MB	~60 min	~5 sec	720×

The speedup grows monotonically from 40 times at 200 KB to 720 times at 5 MB, reflecting the fact that Wi-Fi Direct transfer time grows slowly with file size while NFC transfer time grows linearly. The security overhead of SLDS-E

— comprising NFC mutual authentication, Wi-Fi Direct credential exchange, and network association — remains constant at approximately 4 seconds regardless of file size, as these operations are independent of document size. For large documents, this constant overhead becomes negligible relative to the transfer time itself.

4.10.3 End-to-End Signing Latency

Table 4.4 presents the complete end-to-end timing breakdown for a 500 KB PDF signing session in SLDS-E, comparing each phase against the equivalent SLDS timing.

Table 4.4: SLDS-E End-to-End Timing For A 500 KB PDF Signing Session.

Process	SLDS (sec)	SLDS-E (sec)
Mutual Authentication (NFC)	5.0	2.0
Wi-Fi Direct Credential Exchange (NFC)	—	0.5
Wi-Fi Direct Network Association	—	1.5
File Sending (Signee → Signer)	120.0	2.0
Digital Signing and Embedding	1.5	2.0
File Sending (Signer → Signee)	120.0	2.0
Signature Verification	3.0	0.5
Total	253.5	10.0

The total signing session completes in 10 seconds in SLDS-E compared to 253.5 seconds in SLDS, a 96% reduction. Bidirectional file transfer requires 4 seconds total in SLDS-E compared to 240 seconds in SLDS, as the Wi-Fi Direct channel eliminates the NFC bandwidth bottleneck entirely. SLDS-E is no longer transfer-bound: cryptographic operations now dominate the transaction timeline, which is the intended operational characteristic of a well-designed secure transfer system. The constant security overhead of 4 seconds for authentication and channel setup provides an increasingly favourable security-to-transfer ratio for larger documents, making SLDS-E practical for the full range of professional document sizes.

4.11 Application Screenshots

Figure 4.3 shows the SLDS-E Android Signee application interface, displaying the NFC ready state and active Wi-Fi Direct Group Owner network. Figure 4.4 shows the RSA-SHA256 digital signature embedded in the signed PDF metadata as verified in Adobe Acrobat; the validity warning reflects the use of a self-signed CA certificate.



Figure 4.3: SLDS-E Android Signee application interface.

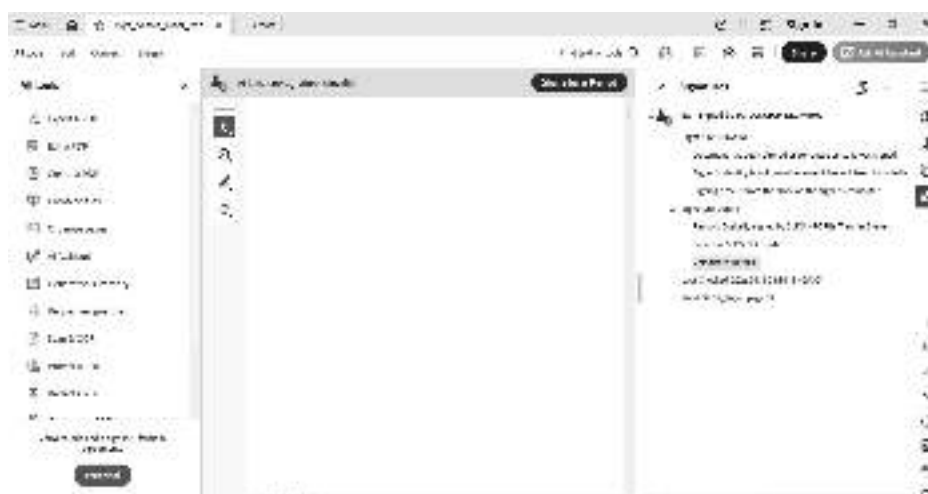


Figure 4.4: RSA-SHA256 signature embedded in signed PDF metadata verified in Adobe Acrobat.

4.12 Use Case: Healthcare Insurance Prescription Signing

The healthcare use case presented for SLDS in Section 3.12 demonstrated the value of proximity-enforced digital signing for prescription authentication and insurance fraud prevention. However, SLDS's transfer time of over 60 minutes for a 5 MB patient record — comprising high-resolution diagnostic imaging, multi-page lab reports, and detailed treatment histories — renders it operationally infeasible for time-critical medical workflows. SLDS-E resolves this critical limitation.

In the SLDS-E workflow, when a physician must digitally sign a prescription or treatment authorization, the patient presents their mobile device for an NFC tap at the physician's workstation PC/SC reader. Mutual authentication completes in 2 seconds, establishing cryptographic proof of physical co-location. Wi-Fi Direct credentials are delivered over NFC in 0.5 seconds, and the PC joins the Android Wi-Fi Direct network in 1.5 seconds. The complete patient record, regardless of size, transfers over Wi-Fi Direct in seconds: a 5 MB record transfers in approximately 5 seconds compared to over 60 minutes in SLDS, enabling SLDS-E to operate within the time constraints of active clinical workflows. The physician reviews the prescription on the full-screen workstation display, applies an RSA-SHA256 digital signature, and the signed document is returned to the patient's mobile health wallet within a total session time of approximately 10 seconds for typical prescription documents.

Insurance providers verify submitted claims by checking the RSA-SHA256 digital signature against the physician’s X.509 certificate. This confirms the authenticity of the signature and the identity of the signing physician. The locality-of-proof guarantee of SLDS-E, maintained through in-band NFC credential delivery while switching to Wi-Fi Direct for bulk transfer, ensures that only a registered physician device close to the patient’s device could apply the signature at signing time. The cryptographic locality guarantee that we provide here is able to stop typical cases of health insurance fraud like remote or illegitimate prescription writing [20].

Chapter 5

Results and Performance Analysis

5.1 Introduction

Chapter combines the performance analysis on SLDS and SLDS-E framework and provides the summary comparative results on all the main parameters such as cryptographic resource efficiency, transfer throughput, end-to-end signing delay and security overhead. The results quantify the performance improvement achieved by SLDS-E over SLDS and demonstrate that this improvement is obtained without any compromise to the security guarantees established by the foundational SLDS framework.

The analysis is structured as follows. Section 5.2 compares the cryptographic resource requirements of both frameworks. Section 5.3 presents the transfer throughput comparison across document sizes. Section 5.4 presents the end-to-end signing latency breakdown. Section 5.5 analyses the constant security overhead of both frameworks. Section 5.6 presents a qualitative security comparison of both frameworks against existing solutions. Section 5.7 discusses the overall findings and their implications for real-world deployment.

5.2 Cryptographic Resource Comparison

Table 5.1 compares the cryptographic resource requirements per device for SLDS and SLDS-E. Both frameworks share the same RSA-2048 and X.509 certificate infrastructure, as SLDS-E inherits the complete SLDS authentication architecture without modification.

Table 5.1: Cryptographic Resource Comparison Between SLDS And SLDS-E Per Device.

Resource	SLDS (bytes)	SLDS-E (bytes)
RSA Public Key	256	256
RSA Private Key	256	256
X.509 Certificate	1024	1024
Symmetric Session Key	128	32
AES-GCM IV (per packet)	—	12
AES-GCM Authentication Tag (per packet)	—	16
Total per device	1664	1596

Both methods have insignificant overhead in storage space usage, with the total cryptographic resource usage being 1664 bytes for SLDS and 1596 bytes for SLDS-E per node. This represents less than 0.002

5.3 Transfer Throughput Analysis

The most significant performance difference between SLDS and SLDS-E lies in document transfer throughput. Table 5.2 presents measured transfer times for both frameworks across four representative document sizes, and Figure 5.1 visualizes the dramatic reduction achieved by SLDS-E.

Table 5.2: Transfer Time And Speedup Comparison Between SLDS And SLDS-E Across Document Sizes.

File Size	SLDS — NFC	SLDS-E — Wi-Fi Direct	Speedup
200 KB	~2 min	~3 sec	40×
500 KB	~6 min	~3 sec	120×
1 MB	~12 min	~5 sec	144×
5 MB	~60 min	~5 sec	720×

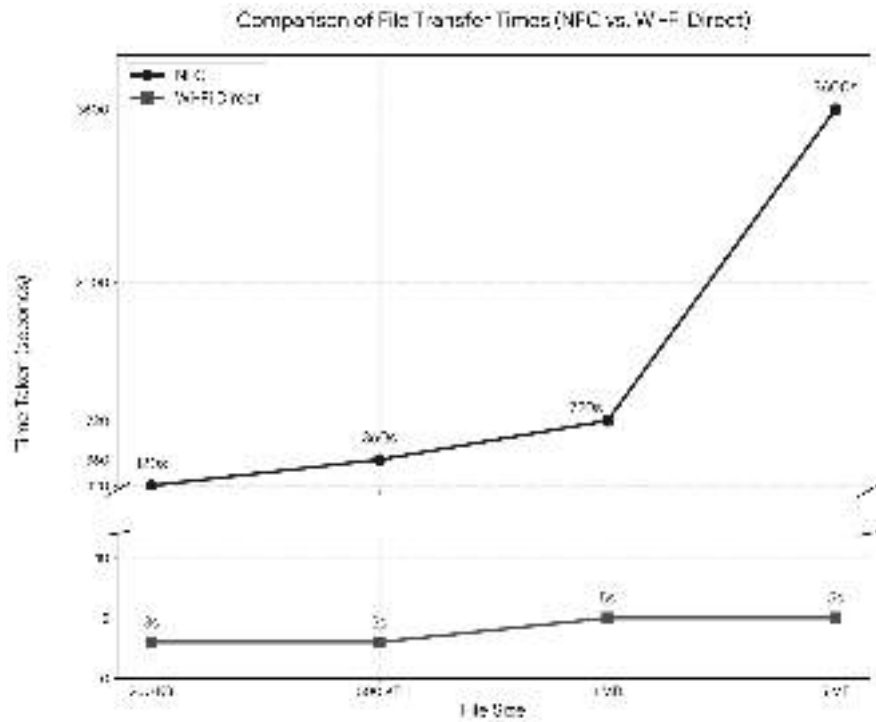


Figure 5.1: Transfer time comparison between SLDS and SLDS-E across document sizes from 200 KB to 5 MB.

It can be noticed that the increase in document size linearly affects the transfer time of SLDS; such dependency stems from the natural restriction of the communication speed of NFC which varies from 106 to 424 kbps. However, transfer time of SLDS-E appears rather stable in a range between 3 and 5 seconds independently of the size of documents tested from 200 KB up to 5 MB; the reason behind this phenomenon is the fact that the average throughput of Wi-Fi Direct equals 2.72 MBps, therefore making any document less than 5 MB insignificant compared to constant time necessary for executing the encryption procedures. The speed-up effect demonstrated by SLDS-E becomes greater and greater along with increasing document size, growing from approximate 40 to around 720 times depending on the size of the transferred file from 200 KB to 5 MB.

A particularly important implication drawn from this comparison is that SLDS-E shows document-size-agnostic transfer speed above a certain limit: after the size of the file exceeds about 200 KB, any further increase in the file size causes almost no change in SLDS-E session time due to the low latency of transferring additional information via Wi-Fi Direct channel. Such property makes SLDS-E applicable for transferring any professional documents – from

a one-page prescription to a multi-MB contract.

5.4 End-to-End Signing Latency

Table 5.3 presents the complete phase-by-phase timing breakdown for a 500 KB PDF signing session in both frameworks, providing a detailed view of where time is spent in each system.

Table 5.3: Phase-By-Phase Timing Comparison For A 500 KB PDF Signing Session In SLDS And SLDS-E.

Process	SLDS (sec)	SLDS-E (sec)
Mutual Authentication (NFC)	5.0	2.0
Wi-Fi Direct Credential Exchange (NFC)	—	0.5
Wi-Fi Direct Network Association	—	1.5
File Sending (Signee → Signer)	120.0	2.0
Digital Signing and Embedding	1.5	2.0
File Sending (Signer → Signee)	120.0	2.0
Signature Verification	3.0	0.5
Total	253.5	10.0
Transfer time as % of total	94.7%	40.0%
Security overhead as % of total	5.3%	60.0%

Several important observations arise from this breakdown. In SLDS, bidirectional NFC file transfer consumes 240 seconds out of 253.5 seconds total, representing 94.7% of the transaction time, with all security operations contributing only 13.5 seconds of overhead. The system is overwhelmingly transfer-bound, meaning that improvements to the cryptographic operations would have negligible impact on the user experience.

In SLDS-E, the total session time of 10 seconds is distributed more evenly across phases, with transfer operations consuming 4 seconds (40%) and security operations consuming 6 seconds (60%). This redistribution is the intended operational characteristic of a well-designed secure transfer system: cryptographic security operations, not data transfer, should be the dominant factor in the transaction timeline. The 2-second improvement in mutual

authentication time between SLDS and SLDS-E reflects implementation optimizations in the SLDS-E authentication phase rather than any protocol change, as the underlying SPDL-specified protocol is identical.

The total reduction from 253.5 seconds to 10 seconds represents a 96% decrease in end-to-end signing latency, transforming SLDS-E from a system requiring over four minutes per signing transaction into one that completes in the time required for a brief user interaction.

5.5 Security Overhead Analysis

The value to both framework that should be underlined most is that security overhead is fixed, regardless of document size, composed from mutual authentication, session key agreement and document integrity verification. Table 5.4 presents the constant security overhead components for both frameworks.

Table 5.4: Constant Security Overhead Comparison Between SLDS And SLDS-E, Independent Of Document Size.

Security Operation	SLDS (sec)	SLDS-E (sec)
Mutual Authentication	5.0	2.0
Wi-Fi Direct Credential Exchange	—	0.5
Wi-Fi Direct Network Association	—	1.5
Digital Signing and Embedding	1.5	2.0
Signature Verification	3.0	0.5
File Metadata Transfer	4.0	—
Total Security Overhead	13.5	6.5

The constant security overhead of 13.5 seconds in SLDS and 6.5 seconds in SLDS-E means that both frameworks provide an increasingly favourable security-to-transfer ratio as document size grows. For a 5 MB document, SLDS’s 13.5-second security overhead represents only 0.37% of the total 3600-second transaction time, while SLDS-E’s 6.5-second overhead represents 56.5% of the total 11.5-second transaction time. This inversion confirms that SLDS-E has successfully eliminated the transfer bottleneck: in SLDS-E, security operations are the dominant factor, which is the correct design outcome for a security-first signing framework.

5.6 Qualitative Security Comparison

Table 5.5 presents a qualitative security comparison of SLDS and SLDS-E against the existing solutions reviewed in Chapter 3, evaluating each system against eight key security and functionality properties.

Table 5.5: Qualitative Security And Functionality Comparison Of SLDS and SLDS-E Against Existing Solutions.

Scheme	Mutual Auth.	MiTM Resist.	Relay Resist.	Replay Resist.	Locality Proof	PC Signer	Formal Verif.	High Throughput
Ponnapalli & Saxena [16]	×	×	×	×	×	✓	×	✓
Weihua & Shuang [17]	×	×	×	×	×	✓	×	✓
iDSign [6]	✓	✓	✓	✓	✓	×	×	×
SecureXfer [7]	✓	✓	✓	✓	✓	×	×	×
SLDS	✓	✓	✓	✓	✓	✓	✓	×
SLDS-E	✓	✓	✓	✓	✓	✓	✓	✓

The qualitative comparison confirms that SLDS is the first framework to simultaneously satisfy mutual authentication, MiTM resistance, relay attack resistance, replay resistance, locality proof, PC-based signing, and formal protocol verification. SLDS-E extends this by additionally providing high throughput, making it the only framework in the literature to satisfy all eight properties simultaneously.

Remote signing solutions such as those of Ponnapalli and Saxena [16] and Weihua and Shuang [17] support PC-based signing and achieve high throughput over network connections, but provide none of the proximity security properties. NFC-based solutions such as iDSign [6] and SecureXfer [7] provide strong proximity security guarantees but are constrained to mobile-to-mobile scenarios and cannot achieve practical throughput for real-world document sizes. SLDS and SLDS-E resolve all of these limitations within a single unified framework.

5.7 Discussion

The experimental results lead to several conclusions regarding the design trade-offs between SLDS and SLDS-E and their suitability for different deployment scenarios.

When SLDS is sufficient: SLDS is suitable for use cases that involve very small documents, generally less than 100 KB, where the duration of NFC

transfer stays within an acceptable range. Typical examples include single-page authorization forms, concise legal affidavits, and prescription documents without any attachments. In such contexts, SLDS benefits from a simpler architecture, as it relies solely on NFC communication and eliminates the need for a Wi-Fi Direct setup phase, thereby reducing the complexity involved in implementation and maintenance. Given the fixed security overhead of 13.5 seconds, SLDS can potentially deliver total session durations comparable to SLDS-E when handling very small documents.

When SLDS-E is necessary: The use of the SLDS-E will be inevitable in the case where there are more than 200 KB of data to be transferred, which includes most of the cases associated with the signing of official documents. Cases such as medical documents involving images, complex legal contracts, documents for auditing purposes, and technical specifications usually contain more information. In such cases, the use of SLDS is associated with a long period of time to transfer the data, making the system non-functional. On the other hand, SLDS-E helps in reducing the latency period by up to 96%.

Security equivalence: The equivalence of security properties between SLDS and SLDS-E warrants particular attention. The enhancements in throughput offered by SLDS-E do not compromise any of the security assurances provided by SLDS. Proof assurance locality still holds as the transmission of credentials takes place using in-band NFC. Identical methods of mutual authentication are present for both architectures. In addition, session key continuity prevents cryptographic problems from occurring should a new key be used in the transmission channel. SLDS-E's increase in performance comes only from its separation of authentication and transmission and does not add to any security flaws.

Practical deployment considerations: Wi-Fi Direct session negotiation increases SLDS-E's session time by only 2 seconds over a hypothetical SLDS with an instant file transfer. This delay is negligible in practice and is offset by removing the NFC bottleneck to any document larger than a few kilobytes. Wi-Fi Direct network setup (Android Signee's activation of the Group Owner network and TCP server) is performed prior to the tap so does not affect the session time that is apparent to the user.

Chapter 6

Conclusion, Future Scope and Social Impact

6.1 Conclusion

This thesis has presented the design, implementation, security analysis, and experimental evaluation of two complementary proximity-based digital signature frameworks — Secure Location DigiSign (SLDS) and Secure Location DigiSign Enhanced (SLDS-E) — developed in progression to address the fundamental limitations of existing digital signing solutions for professional enterprise and healthcare environments.

The motivation for this work arose from three converging requirements that no existing solution simultaneously addressed: the need for cryptographic proof of physical co-location between the signing parties, the need for PC-based large-screen document review before signing, and the need for practical document transfer throughput for real-world file sizes. Remote signing architectures satisfy the throughput requirement but provide no locality guarantee and introduce network-layer vulnerabilities. Existing NFC-based proximity frameworks provide locality enforcement but are constrained to mobile-to-mobile scenarios and cannot achieve practical throughput beyond a few hundred kilobytes. No prior framework addressed all three requirements within a single unified system.

The SLDS framework, presented in Chapter 3, addresses the first two requirements through a certificate-based NFC digital signature architecture enabling a mobile-based Android Signee, operating in Host Card Emulation mode, to tap a PC/SC reader connected to a PC-based Signer and digitally sign a document over NFC. SLDS utilize RSA-2048 key pairs and X.509 certificates issued by the Certificate Authority server which are stored

on hardware-secured tamper-resistance Secure Elements on the involved devices so as to provide hardware-based protection of the private key materials. The formal verification of the unidirectional challenge-response based mutual authentication protocol using the Scyther tool under the Dolev-Yao adversary model proves resistance to man-in-the-middle, replay, impersonation and relay attack for infinite concurrent sessions. The performance of each cryptographic operation has been measured to be of constant time 13.5s independent of document size; and the digital signature generation for RSA-SHA256 has been measured to be of time 1.5s. SLDS shows that the proximity-enforced PC-based digital signature generation with formal security assurance is feasible in practice on commodity hardware without any additional hardware equipment.

Nevertheless, SLDS only use NFC to transfer document, then it would take 253.5 seconds for one signing transaction with a document of 500KB, or more than 60 minutes for a 5MB document, which made the system not workable for any professional real documents.

The SLDS-E framework, presented in Chapter 4, resolves this bottleneck by decoupling mutual authentication from bulk document transfer. SLDS-E retains NFC exclusively for proximity-enforced mutual authentication and Wi-Fi Direct credential exchange, preserving the complete security foundation of SLDS without any modification, while delegating all document transfer to an AES-256-GCM secured TCP channel over Wi-Fi Direct using the NFC-negotiated session key. The critical architectural insight of SLDS-E is the in-band delivery of Wi-Fi Direct credentials exclusively over the authenticated NFC channel, binding the high-speed transfer session to the physical NFC tap event and preserving the locality-of-proof guarantee across the channel transition. Physical co-location within 4–10 cm therefore remains a mandatory prerequisite for initiating document transfer, regardless of whether that transfer subsequently occurs over NFC or Wi-Fi Direct.

Experimental evaluation confirms throughput improvements of 40 times to 720 times over pure NFC transfer across document sizes from 200 KB to 5 MB, reducing a 500 KB signing session from 253.5 seconds to 10 seconds — a 96% reduction in end-to-end latency. Security analysis confirms that SLDS-E inherits all security properties of SLDS and introduces no new attack surfaces. The dual-layer protection of WPA2 link-layer encryption and AES-256-GCM application-layer encryption ensures that compromise of the wireless channel alone cannot recover plaintext document data without the SE-protected session key. SLDS-E is the first framework in the literature to simultaneously provide mutual authentication, MiTM resistance, relay attack resistance, replay resistance, locality proof, PC-based signing, formal protocol verification, and practical transfer throughput for real-world document

sizes.

The healthcare insurance prescription signing use case, presented for both frameworks, demonstrates the practical significance of these contributions. SLDS-E enables a physician to complete a full document signing transaction — including NFC tap, mutual authentication, document transfer, full-screen review, digital signing, and return of the signed document — within approximately 10 seconds, compared to over 60 minutes in SLDS for multi-megabyte patient records. This has moved the system from an academic proof of concept, to a deployable solution in a time-sensitive professional environment.

In summary, this thesis makes the following key contributions to the field of proximity-based digital document signing:

1. A novel certificate-based NFC digital signature framework (SLDS) providing proximity-enforced mutual authentication, PC-based signing, and formally verified security for the first time in a single system.
2. A complete Scyther formal verification of the SLDS mutual authentication protocol under the Dolev-Yao adversary model, providing machine-checked assurance of protocol correctness across unbounded sessions.
3. A hybrid proximity signing framework (SLDS-E) that resolves the NFC bandwidth bottleneck through in-band Wi-Fi Direct credential delivery, achieving 40 times to 720 times throughput improvement while preserving all SLDS security guarantees.
4. Quantitative experimental evidence that the hybrid NFC and Wi-Fi Direct architecture is practically viable for enterprise and healthcare deployment, with a 96% reduction in end-to-end signing latency for a 500 KB document.
5. A demonstration that end-to-end session key continuity across heterogeneous communication channels is achievable without any additional key negotiation step, by reusing the NFC-negotiated AES-256-GCM session key directly for Wi-Fi Direct TCP encryption.

6.2 Limitations

While SLDS and SLDS-E represent significant advances over existing proximity signing frameworks, several limitations of the current implementations should be acknowledged.

Single-Session NFC Constraint: Both frameworks necessitate the continuous presence of the NFC field during the mutual authentication process. Should the NFC field be disrupted—such as when the devices are separated—the session must be restarted from the beginning. In the case of SLDS-E, the NFC field is needed solely for the authentication and credential exchange steps. After these stages, the devices may be moved apart, which reduces this limitation during the transfer phase, although the constraint remains applicable throughout the authentication phase itself.

CA Server Availability for Registration: During the initial phase of device registration, both frameworks necessitate network connectivity to the CA server. Although the signing process operates offline after this initial registration, enterprise environments need to guarantee the CA server’s accessibility and maintain its security. Should the CA server be compromised, the trust model underpinning the entire framework would be jeopardized, given that the legitimacy of all device certificates depends on the CA’s signature.

Single Signer per Session: The current implementations support a single Signer and single Signee per NFC session. Workflows requiring multiple signatories — such as multi-party legal contracts requiring signatures from several authorized parties — would require sequential signing sessions, one per Signer, rather than a single consolidated signing event.

Android and Windows Specificity: The current SLDS-E implementation is specific to Android on the Signee side and Windows on the Signer side, due to the use of the Android HCE API and the Windows Native WiFi API respectively. Extension to iOS Signee devices or Linux/macOS Signer workstations would require platform-specific implementation of the HCE and WiFi Direct components.

NFC Transfer for Small Documents: For documents smaller than approximately 50 KB, SLDS’s NFC-only approach may be operationally adequate and architecturally simpler than SLDS-E. The Wi-Fi Direct setup phase of SLDS-E adds approximately 2 seconds of overhead that is negligible for large documents but represents a proportionally larger fraction of the total session time for very small documents.

6.3 Future Work

The works presented in this thesis suggest several areas of future work and improvement.

Use of Blockchain for Immutable audit trails: The SLDS and SLDS-E documents contain an embedded RSA-SHA256 signature that can

be verified locally using the Signer’s public key. These features can be extended to produce a hash of the signed document on a blockchain during the act of signing. An immovable and tamper evident audit trail can be created this way. Third parties can thereby confirm that the signing event took place without being given access to the original devices. This will be particularly useful in legal and regulatory environments, where evidence of the custody chain is typically required for auditors.

Multi-party proximity signing: Extending SLDS-E for multiple sequential signers in one signing session, and in parallel across several signers on a single Wi-Fi Direct network will help the framework support business processes where a greater degree of complexity is involved, for instance, signing a contract with multiple parties and a board-level approval of a document. The locality-of-proof assurance would most likely extend to this feature, providing a cryptographic proof that all parties are physically present during the signing operation.

Incorporating Post-Quantum Cryptography: Allowing multiple signatories to sign in one signing session, or many signers to simultaneously sign a document via a single Wi-Fi Direct network could enhance the utility of SLDS-E for more complex business processes. These processes range from multi-party contract signing, to board-level document approval, among others. While RSA-2048 is currently a viable encryption method, it is not quantum resistant from Shor’s algorithm, hence future iterations might replace its reliance on mutual authentication and digital signatures with a quantum-resistant scheme (post-quantum) like ML-DSA (Module Lattice-based Digital Signature Algorithm), formerly known as CRYSTALS-Dilithium) and ML-KEM(for key encapsulation) standards that NIST has adopted, in place of RSA-based mutual authentication and digital signature processes. The locality-of-proof guarantee would carry over in this scenario to offer a cryptographic proof that all the parties involved are physically present at the signing session.

Biometric Authentication for Increased Assurance: The current mutual authentication protocol only provides authentication based on the RSA keys. While it authenticates the device, it does not authenticates the individual using the device. It may be useful to incorporate some form of biometric verification, for instance, fingerprint or facial recognition, in the Android Signee application, so as to provide an additional level of authentication and more confidence that the owner of the registered device is indeed the authorized signer. This is crucial in high security scenarios like signing pharmaceutical prescriptions and other financial documents.

Expansion to iOS and cross-platform capability: Supporting iOS clients for the Android Signee application would make the framework usable

on a vast percentage of devices used in both the enterprise and healthcare sector. Since the Core NFC framework support was introduced to iOS in version 11 of the operating system, allowing NFC capabilities to be used including Host Card Emulation through APDU based communication it is possible to port the client application for the iOS platform as well.

Smart Card Integration: Extending the Signer component to accept physical smart card credentials in addition to PC/SC compatible NFC readers will allow the system to interoperate with enterprise smart card systems. A doctor, using a smart card holding their RSA credentials could for instance present it with the patient's Android mobile phone, to a dual-interface reader to effect a 3-way authenticated signature without the need of a PC.

Compression of NFC data: While it may not be desirable to move from SLDS to the more feature-rich SLDS-E for all signing scenarios, compressing the document before sending it over NFC may help alleviate some of the bandwidth limitations on NFC, particularly when dealing with documents that compress effectively. While this is no a solution to the fundamental physical layer limitations of NFC it may serve to reduce transfer time for text-heavy PDF documents by 2-5 times, thereby extending the range of documents that the SLDS framework can realistically handle.

Optimization of NFC throughput: The current TCP implementation in SLDS-E is based on a single TCP transfer stream. Further work could focus on using multiple TCP streams, optimising the packet size to accommodate the MTU on Wi-Fi Direct, and using dynamic chunking algorithms to improve the throughput over the Wi-Fi Direct link to more closely approximate the theoretical maximum for large file transfer over such a network.

6.3.1 Social Impact

Healthcare insurance fraud cost over \$100 billion to global healthcare systems on an annual basis through forgery of treatment documentation, submission of false claims, forged prescriptions and doctored records. At the root of healthcare insurance fraud is the inability to establish a cryptographically verifiable association between the signing physician, the patient being present in physical co-location, and the document being signed. A digital signature implemented at the time of clinical encounter, precisely establishes this cryptographically verifiable association creating an auditable document, the validity of which any insurance carrier could independently verify without reference to external data sources or constant connectivity.

The SLDS-E framework extends upon the existing NFC based health folder architecture by implementing patient-centric health wallet functionality where the patient stores and carries a comprehensive digital medi-

cal record (including treatment orders, lab reports, diagnoses records, allergy lists, medications, etc.) on an NFC-enabled personal mobile device. This translates the patient's portable and cryptographically protected mobile phone into an electronic medical record that the patient carries along at all times, independent of hospitals' central database infrastructure.

In routine usage where physician signing a prescription or any other treatment form would require the doctor to authenticate, the patient will tap her mobile phone against physician's workstation PC/SC reader and the entire SLDS-E four phase process will automatically execute where Mutual Certificate Based Authentication cryptographically verifies the identity of the registered physician and its physical co-location with the patient, exchanged WFD credentials via NFC link, transfer of prescription to physician's PC via fast Wi-Fi Direct for full-screen inspection, the signing physician will review it and append a RSA-SHA256 digital signature and the signed prescription will be transferred back into patient's mobile health wallet, the whole transaction may be finished in around 10 seconds with average prescription document.

In the case of emergency situations where the patients are in unconscious or incapable state to be involved, the doctors who are at the scene will just tap the patients' mobile device for crucial records such as allergic information, prescribed drugs, blood group, known illnesses and prescribed treatment, etc. The locality-of-proof ensured by the proximity of NFC implies that the interaction has to be within 4-10 cm distance, thus no unauthorized device can fetch medical information remotely in this uncontrolled situation; all these sensitive medical information are cryptographically signed with RSA-SHA256 using locality of proof mechanism at the patient's point-of-care with the physically available physician device.

Most of the current medical documents are several mega-bytes in size due to the high-resolution diagnostic imaging or multi-pages of laboratory and treatment documents. For example, a 5 Mega Bytes of patient record could take over 1 hour for NFC transfer that make it infeasible in emergency or operating rooms or out-patient consultation room where doctor is always time limited; the SLDS-E framework supports transferring of this big file over high speed Wi-Fi Direct in less than 5 seconds while all the security guarantees (Mutual Authentication, Locality of proof and non-repudiation) from SLDS are kept same.

The submitted insurance claims will then be validated through by extracting the RSA-SHA256 digital signature out of prescription document and checking with the signing physician's X.509 certificate to verify three main points: authenticity of signature (the document was signed by no one else than by that specific physician), integrity of the document (it was not modified since the signing time), non-repudiation of signing event (the physician

cannot deny his signing intention). In addition, due to the locality-of-proof guarantee, only a registered physician device that is physically available with the patient could have signing that could stop most of the major insurance fraud vector such as remote prescription generation, post-hoc document manipulation and physician identity stealing.

References

- [1] N. A. Kumar *et al.*, “Two key signature scheme with application to digital certificates,” in *Proceedings of the 1st International Conference on Recent Advances in Information Technology (RAIT)*. IEEE, 2012.
- [2] A. Habeeb *et al.*, “Blockchain technology in digital certificate authentication,” in *Proceedings of the 10th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO)*. IEEE, 2022.
- [3] S. Fugkeaw *et al.*, “Proxy-assisted digital signing scheme for mobile cloud computing,” in *Proceedings of the 13th International Conference on Knowledge and Smart Technology (KST)*. IEEE, 2021.
- [4] F. Pfeiffer *et al.*, “Theoretical limits of ISO/IEC 14443 type A RFID eavesdropping attacks,” in *Proceedings of Smart SysTech 2012; European Conference on Smart Objects, Systems and Technologies*. Osnabrueck, Germany: VDE, 2012.
- [5] D. Sethia *et al.*, “NFC based secure mobile healthcare system,” in *Proceedings of the 6th International Conference on Communication Systems and Networks (COMSNETS)*. IEEE, 2014, pp. 1–6.
- [6] H. Sharma and D. Sethia, “iDSign: Secure lightweight digital document signatures framework with mutual authentication and identity-based cryptography,” in *Proceedings of the 6th International Conference on Communication and Intelligent Systems (ICCIS)*. Bhopal, India: IEEE, 2024.
- [7] H. Jaiswal *et al.*, “SecureXfer: Secure NFC RSA mobile-based framework for digital signing of documents,” in *Proceedings of Next Generation Innovations in Smart Environments (NGISE)*. IEEE, 2025.
- [8] C. He, “Research of web resources protection based on digital watermarking and digital signature,” in *Proceedings of the International Con-*

- ference on Intelligent Networking and Collaborative Systems*. IEEE, 2016.
- [9] X. JunWu and X. Fang, “Developing smart card application with PC/SC,” in *Proceedings of the International Conference on Internet Computing and Information Services*. Wuhan, China: IEEE, 2011.
 - [10] A. Zavala *et al.*, “Application to manage digital certificates as a certificate authority (CA) according to the digital signature law of El Salvador,” in *Proceedings of the IEEE 40th Central America and Panama Convention (CONCAPAN)*. IEEE, 2022.
 - [11] A. Alrawais *et al.*, “X.509 check: A tool to check the safety and security of digital certificates,” in *Proceedings of the International Conference on Identification, Information, and Knowledge in the Internet of Things (IIKI)*. IEEE, 2015.
 - [12] D. Sethia *et al.*, “NFC secure element-based mutual authentication and attestation for IoT access,” *IEEE Transactions on Consumer Electronics*, vol. 64, no. 4, pp. 470–479, 2018.
 - [13] T. S. Fun *et al.*, “Enhancing electronic document security with lightweight digital signature,” in *Proceedings of the IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*. IEEE, 2024.
 - [14] C. J. F. Cremers, “The Scyther tool: Verification, falsification, and analysis of security protocols,” in *Computer Aided Verification (CAV 2008)*, ser. Lecture Notes in Computer Science, vol. 5123. Springer, 2008, pp. 414–418.
 - [15] X. Xu *et al.*, “Research on semantics and algorithm of formal analysis tool Scyther,” in *Proceedings of the IEEE 4th International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*. IEEE, 2022, pp. 1058–1074.
 - [16] H. K. B. Ponnappalli and A. Saxena, “A digital signature architecture for web apps,” vol. 15, no. 2, 2013, pp. 38–43.
 - [17] X. Weihua and F. Shuang, “An digital signature method applied for distributed rendering submit system,” in *Proceedings of the IEEE International Conference on Information Science (ICIS)*. Wuhan, China: IEEE, 2017.

- [18] B. D. Yulianto *et al.*, “Digital certificate authentication with three-level cryptography (SHA-256, DSA, 3DES),” in *Proceedings of the International Seminar on Application for Technology of Information and Communication (iSemantic)*. IEEE, 2022.
- [19] N. Kishore *et al.*, “Fast implementation of digital signatures using parallel techniques,” in *Proceedings of the International Conference on Computing, Communication and Green Engineering (CCGE)*. IEEE, 2021.
- [20] M. A. Amin *et al.*, “Utilizing blockchain and smart contracts for enhanced fraud prevention and minimization in health insurance through multi-signature claim processing,” in *Proceedings of the International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*. IEEE, 2024.
- [21] D. Sethia *et al.*, “Smart health record management with secure NFC-enabled mobile devices,” *Smart Health*, vol. 13, p. 100063, 2019.
- [22] M. Jain and D. Sethia, “Secure location DigiSign — A certificate-based NFC digital signature system with PC signer and mobile-based document signee,” in *Proceedings of the 8th International Conference on Innovations in Recent Computational Applications (ICIRCA)*. IEEE, 2026, manuscript submitted for publication.
- [23] U. Demir *et al.*, “Maintaining connectivity in ad hoc networks through WiFi Direct,” in *Proceedings of the IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 2017, pp. 308–316.

APPENDIX I

LIST OF PUBLICATIONS

The following research papers have been accepted for publication based on the work carried out in this dissertation:

1. **Submission ID: 1453**

Title: “Secure Location DigiSign - A Certificate-Based NFC Digital Signature System with PC Signer and Mobile-Based Document Signee”

Conference: 7th International Conference on Data Analytics & Management (ICDAM-2026)

Organized By: London Metropolitan University (UK)

In association with WSG University Portalegre Polytechnic University, SGGW Management Institute

Date: 12–14 June 2026

2. **Submission ID: 1429**

Title: “Secure Location DigiSign with Wi-Fi Direct - A Hybrid NFC Authentication Application for High-Speed Proximity Digital Signing”

Conference: 7th International Conference on Data Analytics & Management (ICDAM-2026)

Organized By: London Metropolitan University (UK)

In association with WSG University Portalegre Polytechnic University, SGGW Management Institute

Date: 12–14 June 2026

I.1 Paper Acceptance Proofs

I.1.1 First Conference Acceptance Proof

5/21/26, 10:09 PM

Gmail - ICDAM 2026: Paper Notification for Paper ID 1453



MUDIT JAIN <jain.mudit100@gmail.com>

ICDAM 2026: Paper Notification for Paper ID 1453

8 messages

ICDAM 2026 <icdam.conf@gmail.com>
 To: Jain Mudit <jain.mudit100@gmail.com>

Thu, May 14, 2026 at 2:39 PM

7th International Conference on Data Analytics & Management (ICDAM-2026)!

Dear Author(s),

Greetings from **ICDAM 2026**!

We congratulate you that your paper with submission ID **1453** and Paper Title '**Secure Location DigiSign - A Certificate-Based NFC Digital Signature System with PC Signer and Mobile-Based Document Signee**' has been accepted for publication in the Springer LNNS series [Approved]-[Indexing: SCOPUS, INSPEC, WTI Frankfurt eG, zbMATH, SCImago; All books published in the series are submitted for consideration in Web of Science]. This acceptance means your paper is among the top 20% of the papers received/reviewed. We urge you to complete your registration immediately to secure your spot at this highly anticipated event. **Kindly complete the registration by 17th May 2026. No further Extension.**

Please register as soon as possible and submit the following documents to icdam.conf@gmail.com as soon as possible.

1. Final Camera-Ready Copy (CRC) as per the springer format. (See <https://icdam-conf.com/downloads>)
2. Copy of e-receipt of registration fees. (For Registration, see <https://icdam-conf.com/registrations>)
3. The final revised copy of your paper must be mailed to the conference mail id.

Note : Standard Paper size – 10-12 pages. Over length of more than 12 pages, paper charges USD 20 per extra page

The reviewers comments are given at the bottom of this letter, please improve your paper as per the reviewers comments. While preparing the final CRC manuscript, kindly check the following google link of proceedings of the previous International Conference on Data Analytics and Management:

<https://link.springer.com/conference/icdam>

and it is suggested to cite the relevant latest papers matching the area of your current research paper.

<https://mail.google.com/mail/u/0/?ik=77b8f31419&view=pt&search=all&permthid=thread-f:1865154310346436881&simpl=msg-f:1865154310346436881...> 1/7

I.1.2 Second Conference Acceptance Proof

5/21/26, 10:12 PM

Gmail - ICDAM 2026: Paper Notification for Paper ID 1429



MUDIT JAIN <jain.mudit100@gmail.com>

ICDAM 2026: Paper Notification for Paper ID 1429

9 messages

ICDAM 2026 <icdam.conf@gmail.com>
 To: Jain Mudit <jain.mudit100@gmail.com>

Thu, May 7, 2026 at 12:28 PM

7th International Conference on Data Analytics & Management (ICDAM-2026)!

Dear Author(s),

Greetings from **ICDAM 2026**!

We congratulate you that your paper with submission ID **1429** and Paper Title '**Secure Location DigiSign with Wi-Fi Direct - A Hybrid NFC Authentication Application for High-Speed Proximity Digital Signing**' has been accepted for publication in the Springer LNNS series [Approved]-[Indexing: SCOPUS, INSPEC, WTI Frankfurt eG, zbMATH, SCImago; All books published in the series are submitted for consideration in Web of Science]. This acceptance means your paper is among the top 20% of the papers received/reviewed. We urge you to complete your registration immediately to secure your spot at this highly anticipated event. **Kindly complete the registration by 10th May 2026. No further Extension.**

Please register as soon as possible and submit the following documents to icdam.conf@gmail.com as soon as possible.

1. Final Camera-Ready Copy (CRC) as per the springer format. (See <https://icdam-conf.com/downloads>)
2. Copy of e-receipt of registration fees. (For Registration, see <https://icdam-conf.com/registrations>)
3. The final revised copy of your paper must be mailed to the conference mail id.

Note : Standard Paper size – 10-12 pages. Over length of more than 12 pages, paper charges USD 20 per extra page

The reviewers comments are given at the bottom of this letter, please improve your paper as per the reviewers comments. While preparing the final CRC manuscript, kindly check the following google link of proceedings of the previous International Conference on Data Analytics and Management:

<https://link.springer.com/conference/icdam>

and it is suggested to cite the relevant latest papers matching the area of your current research paper.

I.2 Scopus Indexing Proof

09/05/2026, 16:54

ICDAM



7th International Conference on
Data Analytics & Management
(ICDAM-2026)

Theme: Data Analytics with Computer Networks

Organized By: London Metropolitan University (UK)

In association with WSG University • Portalegre Polytechnic University • SGGW Management Institute

Date: 12–14 June 2026

Springer LNNS (Indexed in Scopus, EI, WoS)



About the Conference

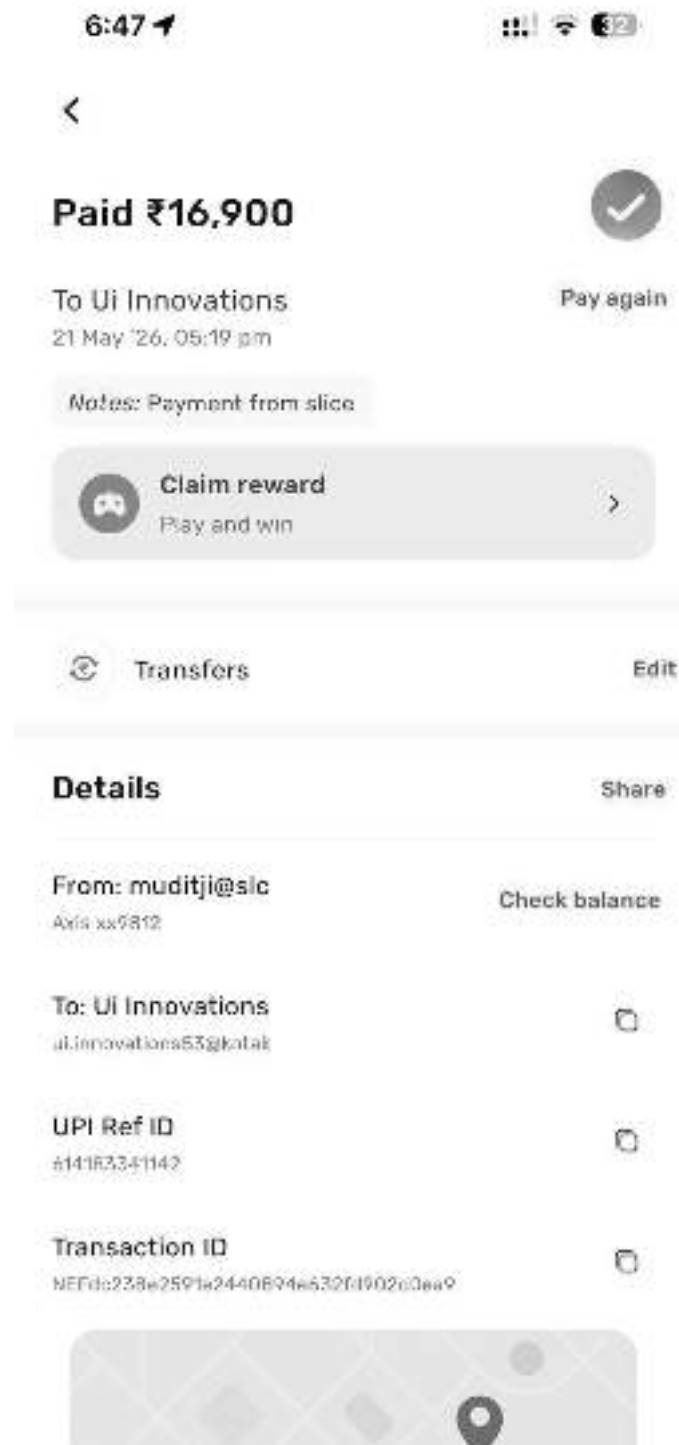
The International Conference on Data Analytics and Management (ICDAM-2026) is organised with the objective of bringing together innovative scientists, professors, research scholars, students and industrial experts in the field of Computing and Communication to a common forum.

The primary goal of the conference is to promote the exchange of innovative scientific information between researchers, developers, engineers, students, and practitioners. Another goal is to promote the transformation of fundamental research into institutional and industrialized research and to convert applied exploration into real time application.

Overall, the conference will provide researchers and attendees with opportunities for national and international collaboration and networking among universities and institutions from India and abroad for promoting research.

I.3 Conference Paper Registration Receipts

I.3.1 First Conference Receipt



I.3.2 Second Conference Receipt





Paid ₹16,900


To Ui Innovations
 10 May '26, 03:13 pm

Pay again

Notes: Payment from slice


Claim reward
 Play and win




Transfers

Edit

Details

Share

From: muditji@slc
 Axis xx9812

Check balance

To: Ui Innovations
 ui.innovations53@kotak



UPI Ref ID
 613C19583919



Transaction ID
 NCF08b10664374f16a1a28ee53c795102c0



Add extra notes

AUTHOR'S PROFILE

Mudit Jain is pursuing his Master of Technology (M.Tech.) degree in Software Engineering from the Department of Software Engineering at Delhi Technological University (DTU), Delhi, India. He currently works as an intern at NXP Semiconductors in Noida. His M.Tech. thesis is titled *Secure Location DigiSign: A Proximity-Enforced NFC Digital Signature Framework with Certificate-Based Authentication and Hybrid Wi-Fi Direct Transfer*. It focuses on certificate-based mutual authentication over NFC, formal protocol verification against adversarial models, and designing hybrid proximity signing frameworks that combine NFC and Wi-Fi Direct for real-world document signing.



During his postgraduate studies, he has worked on designing tamper-resistant proximity signing frameworks that use Secure Elements, implementing mutual challenge-response authentication protocols, and formally verifying these protocols against various adversarial models using the Scyther tool, including the Dolev-Yao model. He also designed a hybrid NFC/Wi-Fi Direct signing architecture that improved throughput significantly, achieving gains from 40x to 720x compared to pure NFC transfer, while maintaining essential security guarantees.

His research has been accepted and published at several international conferences focused on proximity security, NFC-based digital signatures, and hybrid authentication architectures. This includes the design and formal verification of the SLDS mutual authentication protocol, which is proven secure under the Dolev-Yao (unbounded session) model against various attacks such as man-in-the-middle, replay, and impersonation. He also developed the SLDS-E hybrid framework, which sped up the digital signing of a 500KB document from 253.5 seconds to 10 seconds, a 96% improvement, while maintaining proximity security and cryptographic non-repudiation. Both papers have been accepted in Springer's Lecture Notes on Networks and Systems (LNNS) and are indexed in SCOPUS, INSPEC, Web of Science, and EI Compendex. He is also a verified peer reviewer on Web of Science and has reviewed 19 papers across 4 IEEE conferences.