Hiding Patient Data in Medical Images in DICOM Metadata

K. Srinivas, M. Divya*, B. Iswaryya Lakshmi, Arushi Mathur, B.Vijay Daniel

Vignan Institute of Technology and Science, JNTUH, India

srinivas.kmt@gmail.com, muppavaramdivya@gmail.com, iswaryyareddy@gmail.com,

arushimathur2604@gmail.com, anjankumar1612@gmail.com

*K.SRINIVAS (srinivas.kmt@gmail.com)

ABSTRACT

Keeping medical images and patient records in separate systems creates a huge security hole. Central databases are basically magnets for hackers. And if technicians strip out metadata to share a scan safely, doctors lose the clinical context they need. We fixed this structural flaw by locking patient data directly inside the DICOM file. We never touch the image pixels. Even minor pixel changes ruin a scan's diagnostic value. Metadata steganography bypasses that issue entirely. The security setup operates in layers. Patient text undergoes AES-256 encryption. We apply DNA cryptographic mapping next, and then hide the ciphertext inside reserved DICOM tags. The medical scan stays completely intact. During testing, the implementation hit an infinite Peak Signal-to-Noise Ratio (PSNR). That means zero visual distortion. Data extraction works perfectly. We also built in a SHA-256 hash check to catch any tampering instantly. Tying clinical identity straight to the imaging file lets the system ignore vulnerable central storage. It creates a highly secure, self-contained medical record.

Keywords: DICOM, DNA Cryptography, AES Encryption, Steganography, Healthcare Security, Medical image Steganography.

1. Introduction

Medical imaging drives modern healthcare. Doctors rely heavily on MRIs, CTs, and X-rays for diagnoses. These systems almost always export to the DICOM format. It is a highly practical file type because it bundles high-resolution images with text metadata, such as clinical histories. But that exact feature is a privacy nightmare. Storing sensitive data right in the file structure creates major security risks during network transfers.

Most hospitals try to fix this by splitting things up. They put images in one system and lock patient details in a separate database. Even with encryption, those central servers are huge targets for cyberattacks. A single breach exposes thousands of records. Decoupling a scan from its clinical file also causes practical issues. Records inevitably get mismatched. If an image loses its context during a hospital transfer, the doctor lacks the background information needed for a safe diagnosis. Steganography offers an alternative. It hides text inside the digital media itself. The problem is that standard steganography tweaks image pixels. In medical diagnostics, altering even one pixel is unacceptable. Pixel shifts create digital artefacts that look identical to tumours, destroying the scan's accuracy. Any secure data-hiding method must leave the visual image completely untouched. We built a secure hybrid framework to tackle this exact hurdle. It embeds patient info directly into the DICOM file without touching the pixel array. We rely strictly on metadata steganography. First, we secure the patient’s history using AES-256, combined with DNA-based cryptographic principles. Then we inject that ciphertext straight into reserved private DICOM tags. We included a SHA-256 hash check to verify that the extracted data is authentic and untampered.

This approach permanently ties a patient's identity to their scan. We drop the reliance on vulnerable external databases. The image file becomes its own encrypted vault. Only authorised staff with the right cryptographic keys can extract the text. The result is a distortion-free system designed for safely sharing healthcare data.

Proceedings of ICICCS-2026 (Poster Papers), DMPedia Lecture Notes in Computer Science & Engineering.
ISBN: 978-81-993813-1-5

2. Research Methodology

We built a security setup that mixes cryptography with metadata steganography. The goal is straightforward: hide the patient's private details within the DICOM file so no one else can see them, without ruining the actual medical scan. We only put the encrypted data inside the file's metadata tags to make this happen. We never touch the image pixels.

A. System Architecture

We designed the system with two distinct sides for sending and receiving. On the sender side, the program takes a DICOM image and the patient's text. It scans the file's metadata to find non-critical tags. We need a safe spot to hide the data without breaking the file structure. Once we pick a tag, we convert the patient text into binary. We encode that binary using a DNA mapping technique next. We then run that DNA-encoded string through an AES algorithm to lock it down. We also create an RSA digital signature based on that encrypted text to prove the file hasn't been hacked later. We pack the encrypted text and the signature together, compress them, and write the result to the DICOM tag. The final output looks and acts like a normal X-ray. On the receiver side, everything runs in reverse. The software targets the specific metadata tag and extracts the hidden block. It decompresses the block and immediately checks the RSA signature using the public key. If the signature matches, we know the file is safe. The system then decrypts the AES-encrypted data, reverses the DNA mapping, and prints the original text.

B. Algorithm Description

The code handles the data in a very specific sequence. The program reads the DICOM file first to extract the metadata. We convert the patient's information into binary. The DNA encoding part happens next. This maps every pair of binary bits to a nucleotide base, such as A, C, G, or T. It adds a preliminary layer of scrambling before the heavy encryption starts. We take that DNA string and run it through our AES algorithm using a randomly generated secret key. At the same time, we use the sender's private key to build an RSA signature. We tie the signature to the encrypted text. We compress them to keep file sizes down and inject the payload directly into the chosen metadata tag. To read the data later, the receiving software opens the DICOM file and targets that hidden block. It unzips the package to separate the text from the signature. The script verifies the RSA signature before doing anything else. If the signature fails, the whole process stops. Loading corrupted data is a massive clinical risk. Assuming the check passes, we take the shared secret key and unlock the AES layer. The last step involves translating that DNA sequence back into normal text. The patient's information stays completely hidden from unauthorised users. At the same time, the doctor gets an unaltered image.

3. Results and Discussion

Validating the framework required real clinical data. We ran the experiments on a batch of 27,560 microscopic blood smears. We kept the dataset completely even by splitting the infected and healthy cells right down the middle. We applied a standard 80/20 split for training and testing after that.

A. Training and Testing Accuracy

Our 150-layer CNN was built directly on top of MobileNet. We used transfer learning to keep training time manageable. We also applied some basic data augmentation to prevent the network from just memorising the images.

The learning curve stabilised nicely. Training accuracy peaked at 98.72%. The test accuracy followed right behind it at 97.89%. Barely any gap exists between those two numbers. This proves the model avoids overfitting and actually understands the visual features.

B. Performance Comparison with Baseline Models

We ran a standard VGG16 network under the same hardware conditions to act as a benchmark.

VGG16 Baseline: 94.31% Accuracy | 93.84% Precision | 94.12% Recall | 93.98% F1-Score

Our MobileNet (150CNN): 97.89% Accuracy | 97.65% Precision | 98.02% Recall | 97.83% F1-Score

The MobileNet setup easily outperformed the VGG16 baseline across all metrics. It did this while requiring way fewer trainable parameters. You could realistically run this on a standard hospital computer without buying expensive hardware.

C. Confusion Matrix Analysis

Checking the confusion matrix for the test set shows the network rarely trips up. It correctly identified healthy and parasitised cells the vast majority of the time. False positives and false negatives stayed extremely low. Minimising that misclassification rate is the most critical factor in a real clinical setting.

D. Evaluation of the Secure DICOM Framework

Beyond image classification, we tested AES encryption within DICOM headers. The steganography implementation worked exactly as planned. We embedded the encrypted patient text right into the metadata tags. We never touched the pixel array. The medical scans kept their full diagnostic quality. On the extraction side, the software pulled the payload perfectly. The digital signature confirmed that no data was corrupted during storage. We observed zero visual damage to any reconstructed files.

E. Summary of Results

The 150CNN-MobileNet architecture outperforms older networks like VGG16. Relying on dense connections helped us reuse features while keeping the math simple. The DICOM security layer successfully hides records inside the image file without degrading the scan. We managed to put together a system that is lightweight, highly accurate, and genuinely secure enough for clinics.

4. Conclusions

Securing patient records without degrading the actual medical image is a major hurdle in digital medicine. Relying on central databases or basic encryption leaves too many gaps. Those traditional setups essentially act as giant targets for hackers, making routine file transfers a constant privacy risk.What we built proves that combining strong cryptography with metadata steganography works for safe file sharing. Because we embedded the encrypted text straight into the DICOM header, the image pixels were never touched. The scan keeps all of its diagnostic value. During extraction, data recovery was 100% accurate, and integrity checks passed without issue, effectively preventing unauthorised access.Tying the clinical data directly to the image file removes the need for central databases altogether. This drops the biggest single point of failure in a hospital's IT network. Plus, processing the text with DNA-based AES encryption before injecting it makes the payload incredibly secure. If an attacker somehow finds the hidden block, the contents remain completely useless to them.In the end, this system offers a highly practical tool for clinical environments. Doctors can safely transmit files across cloud networks without sacrificing any diagnostic accuracy.

5. Future scope

The current framework already effectively secures standard 2D DICOM files. That is a great baseline, but the immediate next challenge is scaling it up. We need the system to handle full

3D scans and multi-frame images. Think continuous CT volumes or dynamic MRIs. Hiding data across a moving sequence is a lot harder than hiding it in a static image. You have to keep the steganography perfectly synced over time without ruining the diagnostic quality of any single slice.We also want to make the data-hiding process much harder to crack. By bringing in deep learning or texture-aware embedding, the system could adapt on the fly to the specific structure of an image. That would make the payload way more robust against modern steganalysis attacks, which is especially important if hospitals start sharing these files over open networks.Looking further down the road, upgrading the cryptography is a huge priority. Moving to post-quantum encryption or adopting a blockchain-based key management system would massively boost the system's long-term security. It would also make it much easier to audit who is accessing what across different medical institutions.Speed is another major factor. If we optimise the algorithm to run in near real time, it could plug directly into standard hospital PACS or cloud networks. Doctors do not have the time to wait around for sluggish security protocols while making a diagnosis. Legal compliance is the last big hurdle. Privacy rules change constantly. If we bake automated access controls into the framework, the framework will naturally adapt to new healthcare regulations. That level of flexibility is an absolute requirement as hospitals continue to upgrade their digital networks.

References

[1] P. L. K. Mantos and I. Maglogiannis, “Sensitive patient data hiding using a ROI reversible steganography scheme for DICOM images,” *Journal of Medical Systems*, vol. 40, no. 156, pp. 1–17, 2016.

[2] I. Maglogiannis, E. Zafiropoulos, A. Platis, and C. Lambrinoudakis, “Web-based medical image exchange and processing,” *Journal of Medical Systems*, vol. 33, no. 6, pp. 453–470, 2009.

[3] A. Al-Qershi and B. Khoo, “Authentication and data hiding using a hybrid ROI-based watermarking scheme for DICOM images,” *Journal of Digital Imaging*, vol. 24, no. 1, pp. 114–125, 2011.

[4] F. Petitcolas, R. Anderson, and M. Kuhn, “Information hiding—A survey,” *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1062–1078, Jul. 1999.

[5] DICOM Standards Committee, *Digital Imaging and Communications in Medicine (DICOM) Standard*, NEMA, Rosslyn, VA, USA, 2023.

[6] J. Fridrich, “Applications of data hiding in digital images,” in *Proceedings of the IEEE International Conference on Information Technology*, 1998, pp. 1–6.

[7] C.-K. Chan and L. M. Cheng, “Hiding data in images by simple LSB substitution,” *Pattern Recognition*, vol. 37, no. 3, pp. 469–474, 2004.

[8] S. Dube, K. Sharma, “Hybrid approach to enhance contrast of image for forensic investigation using segmented histogram,” International Journal of Advanced Intelligence Paradigms 13, no. 1-2 (2019): 43-66.

[9] X. Zhang and S. Wang, “Efficient steganographic embedding by exploiting modification direction,” *IEEE Communications Letters*, vol. 10, no. 11, pp. 781–783, Nov. 2006.

[10] J. Coatrieux, H. Maitre, B. Sankur, Y. Rolland, and R. Collorec, “Relevance of watermarking in medical imaging,” *IEEE EMBS Magazine*, vol. 19, no. 5, pp. 34–45, Sep.–Oct. 2000.

[11] W. Diffie and M. Hellman, “New directions in cryptography,” *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976.

[12] National Institute of Standards and Technology (NIST), *Advanced Encryption Standard (AES)*, FIPS PUB 197, Nov. 2001.