# Technical Report: PDF Metadata Analyzer for Document Tampering Detection

## 1. Approach, Assumptions, and Methodology

**Approach:**
The PDF Metadata Analyzer detects document tampering using only metadata—without analyzing document content. It identifies inconsistencies in embedded metadata to flag manipulation attempts.

**Key Assumptions:**

1. Tamperers often overlook metadata.
2. Metadata patterns (e.g., suspicious software, illogical dates) suggest forgery.
3. It targets less sophisticated tampering.
4. Metadata analysis is a preliminary screening tool.

**Methodology:**

1. **Extraction:** Use PyPDF2 to extract standard and XMP metadata.
2. **Consistency Checks:** Analyze creation/modification dates, future timestamps, and software legitimacy.
3. **Pattern Matching:** Compare metadata to trusted software lists and required fields.
4. **Integrity Checks:** Assess structural coherence to detect corruption.

## 2. Challenges and Trade-offs

**Technical Challenges:**

- **Metadata Variability:** Differences due to creation tools, workflows, or industry norms can cause false positives.
- **Sophisticated Tampering:** Forgers may generate legitimate-looking metadata.
- **Format Limitations:** PDFs allow optional/custom metadata, complicating analysis.

**Implementation Trade-offs:**

- **Simplicity vs. Comprehensiveness:** Focused on lightweight, fast metadata-only analysis, sacrificing detection depth.
- **False Positives vs. Negatives:** Tuned for high sensitivity, raising more flags with human review required.
- **Generalization vs. Specialization:** Designed for broad use, missing document-specific forgeries.

**3. Improvements and Scaling**

**Enhancements:**

- **Content Analysis:** Add OCR, image, and layout verification.
- **Machine Learning:** Train models on real and tampered documents.
- **Blockchain:** Use distributed ledgers for verifiable provenance.

**Scaling Strategies:**

- **Microservices Architecture:** Modular services with scalable APIs.
- **Institutional Integration:** Partner with institutions for data and verification APIs.
- **Knowledge Base:** Maintain patterns, software fingerprints, and forgery techniques.

**Conclusion**

The tool offers a fast, resource-efficient method for initial forgery detection. While not comprehensive, it lays the groundwork for a multi-layered verification system combining metadata, content analysis, machine learning, and institutional integration.

- SAITEJA CHEKURI