WarrInt: Integrity Validation for Criminal Legal Process

Alisha Ukani
UC San Diego
aukani@ucsd.edu

Ross Greer
UC Merced
rossgreer@ucmerced.edu

Katherine Izhikevich
UC San Diego
kizhikev@ucsd.edu

Earlence Fernandes
UC San Diego
efernandes@ucsd.edu

Alex C. Snoeren
UC San Diego
snoeren@cs.ucsd.edu

Stefan Savage
UC San Diego
savage@cs.ucsd.edu

Abstract

In the U.S., criminal legal process (e.g., a search warrant) is the key mechanism by which law enforcement entities and courts compel third parties (e.g., Google or Meta) to produce evidence in support of a criminal investigation. However, these legal orders are simply documents, commonly served electronically, and there is no inherent mechanism to guarantee their integrity. In recent years, law enforcement has increasingly reported the use of forged legal process used to advance a variety of malign purposes, with online sellers offering such capabilities for a fee. We design a practical system, *WarrInt*, that is compatible with the constraints of existing process mechanics and helps third parties identify when the legal orders they receive may be forged or tampered with. Our evaluation on real and modified legal process documents shows that *WarrInt* is effective at identifying modifications.

1 Introduction

This paper focuses on an underappreciated attack vector on online service providers: legal orders. In the United States, for example, the executive and judicial branches are empowered to demand data of third parties in support of criminal investigations. Thus, large technology companies routinely receive criminal legal process—legal documents demanding specific information pursuant to specific government authorities—such as subpoenas or search warrants. However, the mechanism by which such legal process is created, modified, and served has limited intrinsic integrity protections and, thus, a recipient may not be able to discern if any given legal request is valid or fraudulent. As former Department of Justice attorney Mark Rasch put it, "there's no real mechanism ... [for] tech companies to test the validity of a search warrant or subpoena. And so as long as it looks right, they'll comply" [37].

Unfortunately, this attack vector is being abused in practice; in September 2023, Verizon received a fraudulent search warrant and, as a result, gave the physical address of a woman to her stalker [10]. Others have documented criminal forgeries of civil court orders [7], a court-ordered wiretap request [27], and criminals offering to impersonate law enforcement personnel for $100–250 as part of an online "Warrant/subpoena service" [37]. In 2024, an FBI bulletin similarly documented that criminals were using compromised government accounts to issue forged legal requests [25], further echoed by private-sector claims that "forged court orders and more are all for sale to criminals online" [19,20]. Indeed, concerns about these issues led three U.S. Senators to propose the "Digital Authenticity for Court Orders Act" which would have required some form of digital signature technology for all federal and state court orders [54]. However, the practicalities of criminal legal process creation, authorization, and service complicate the application of traditional digital signature approaches.

Our work is focused on documenting—and developing around—the constraints imposed by existing practice and norms in the U.S. These include requirements for supporting paper process (physical document creation, scanning, and/or delivery are still widely used), compatibility with existing forms and systems (e.g., not requiring changes to existing forms, recording systems, or databases), ease of use (for both court staff and company legal analysts), and the ability to support decentralization and incremental deployment. From these we have designed a prototype system, *WarrInt*, that balances these constraints while offering significant practical improvements in legal process integrity. Testing against existing and modified criminal process documents, we demonstrate that *WarrInt* identifies all attempted forgeries, without unduly misleading analysts into rejecting legitimate process.

This paper offers three principal contributions, including: a) documenting, for the academic community, the nature of how U.S. criminal legal process (particularly Federal criminal process) is created and served on third-party Internet companies and the nature of the risks entailed by the lack of integrity protections, b) identifying concrete security and functionality goals compatible with existing practice and norms, and c) designing, implementing, and evaluating a prototype system to help legal compliance staff detect forged and/or modified legal process subject to these constraints.

2 Background

Governments commonly give themselves special powers to investigate crimes, including the ability to demand information and seize property as needed, via some standardized *legal process*. In this paper, we focus on how this authority manifests in U.S. Federal law and this section serves to review the state of practice—law, mechanism, and norms—by which the Federal government obtains data from third-parties (e.g., email providers, mobile phone carriers, and online social networks). In providing this background, we are informed by a combination of written court procedures, statutory law, and concrete court documents (the collection of which is described further in Section 6), as well as interviews with key stakeholders including current and former Federal law enforcement agents, Assistant U.S. Attorneys, Federal magistrate judges, courtroom clerks, and the legal compliance staff at several of the largest service providers in the U.S.

We have chosen to focus particularly on the U.S. Federal system for three reasons: 1) it is the system with which we have the most experience and familiarity, 2) many of the largest online service providers (e.g., Microsoft, Google, Meta, Apple, Amazon, etc.) are U.S. companies and therefore subject to the Stored Communications Act, which prohibits them from providing content except in response to orders from U.S. courts and thus impacts their ability to comply with foreign law enforcement demands, and 3) the U.S. Federal Courts, via the Public Access to Court Electronic Records (PACER) service, provide public access to an unusual array of criminal process documents (which, in our experience, are rarely available to the public in most other jurisdictions). While we believe that much of this context is shared with other court systems (particularly U.S. state courts), it is neither universal nor inherent, but rather a byproduct of U.S. jurisprudence.

2.1 Common Forms of Criminal Process

U.S. Federal law authorizes, either via the Federal Rules of Criminal Procedure (FRCP) [26] or specific statutes (e.g., the Stored Communication Act), an array of authorities under which the government may obtain information from third parties to support criminal investigations. For retrospective (i.e., stored) data, the three most common forms of process are subpoenas, which can be used to demand basic subscriber records (e.g., identity and billing information); 2703(d) orders, which can reach metadata (e.g., to/from-style records); and search warrants, which are needed to demand the production of content (e.g., the subject and body of an email or DMs in an online social network). We focus on search warrants in this section and describe the requirements and mechanics for other forms of process in Appendix C; all involve a process document authorizing the government to demand certain information or assistance in obtaining such information.

2.2 Process Mechanics

There are three main steps in the life cycle of Federal criminal process execution. First, an authorized process document must be obtained (which may include orders to ensure secrecy), then it must be served upon the entity who possesses the records of interest (e.g., a service provider like Google), and then that entity (if they do not challenge the request[1]) will provide "returns," i.e., data responsive to the request.

2.2.1 Obtaining a Warrant

To obtain a Federal search warrant, the requesting party (typically an Assistant U.S. Attorney) will complete two forms: an application and a proposed warrant that identify both the provider and accounts to be searched.[2] The application will also include a signed affidavit of a law enforcement agent explaining why there is probable cause to believe that the requested search will provide evidence of a crime. In addition, both the application and the proposed warrant will include two attachments: Attachment A, which describes the physical "location to be searched" (i.e., the provider's address) and Attachment B, which describes the "items to be seized" and may include directives to the provider specifying identifiers to search and what databases or services they wish to be included in the search. Search warrant requests are typically handled by Magistrate Judges in each of the 94 Federal districts, who question the agent under oath and determine if the request is consistent with the law—either accepting it, rejecting it or, occasionally, modifying the request to make it compliant. If the request is accepted, the judge will sign and date both the warrant application and the proposed warrant and identify a time frame during which the warrant must be executed.

Once signed by the reviewing judge, the application, search warrant, affidavit, and attachments are filed by court clerks, who assign a case number[3] and, if used, apply a court's stamp or seal to the documents. As a result, this means that documents are routinely modified after a judge signs them. While a court's order is legally binding without the case number and stamp or seal, our interviews reveal that law enforcement prefers to serve the complete, filed version of an order due to a perception that providers will balk at a warrant or other order that looks atypical (i.e., without a case number).[4]

[1] When physical search or seizure is authorized, law enforcement conducts the search, including using force if necessary. Hence, validation by the entity being served plays no role (i.e., the individual has no right to object to the search in the moment). In contrast, in the provider context they are being ordered to execute the search on behalf of the government.

[2] These are commonly based on forms standardized by the Administrative Office of the U.S. Courts, e.g., AO106A, AO93C, etc [58].

[3] Investigatory criminal process is typically not associated with an existing indictment or complaint and, thus, receives its own unique case number which commonly is not known until the matter is heard by a judge [61].

[4] Supporting this understanding, we have yet to encounter a single example of a warrant served on a provider that did not include a valid case number.

While historically Federal warrants were purely paper instruments—signed, dated, and stamped in ink—they are now a mix of ink and digital annotation, recorded as PDFs. Thus, while a draft warrant may originate as a digital document, it may then be printed by the clerk for the judge to sign "wet" (only to be scanned again), or the judge may apply a digital "stamp" signature (i.e., an image of a signature) directly to the PDF.[5] Similarly, the case numbers and any district court stamps or seals added by the clerk may themselves be digital or analog, or some mix of the two. The final warrant document, comprising the signed warrant and its two attachments, is both filed in the court's standard online filing system, Case Management/Electronic Case Files (CM/ECF), and provided to the investigating law enforcement agent (commonly via email), who is then responsible for serving it on the provider. These practices may also vary between the 94 Federal districts, as each operates independently.

A final complication is that, by default, most criminal process documents are filed "under seal"—meaning that they are not available to the public due to their sensitivity [9, 42]. Thus, they do not appear on public dockets and the court may not acknowledge their existence if asked. Furthermore, recent concerns about system security have led many districts to require that such sealed documents be filed on paper, segregated from CM/ECF [49].

2.2.2 Process Service

Historically, all criminal process was served in person, but, over time, the courts and many corporations have come to accept service via certified mail, fax, or the Internet. However, there can be considerable variation in how organizations will accept service.[6] For example, Pitney Bowes (online shipping and e-commerce technology provider) accepts electronic process requests via fax; GitHub and 23andMe accept process via postal mail and email; Block (which operates Square and Cash App services) accepts legal process served via an online Web form; and Electronic Arts requires in-person service.

To better manage the large volume of government data requests, the largest online service providers frequently operate dedicated online portals for this purpose. For example, Google [31], Microsoft [47], and Meta [24, 64] all operate their own portals, while the service Kodex [36] provides an outsourced law enforcement portal service used by many others (including OpenAI, Twilio, Coinbase, Yahoo, and Discord). Such portals require law enforcement officers to first register an account with the system. Registration is particularly sensitive because it requires the portal operator to establish that a registrant is indeed a sworn law enforcement representative. Unfortunately, there is no central repository for the names or identifying information for the 1.2+ million sworn officers operating in the U.S. Thus, each operator has developed proprietary heuristic methods to help validate that any new account is actually associated with an authorized sworn agent (e.g., via their use of known law enforcement email domains such as `fbi.gov`). Once an account exists, an officer will authenticate, then upload process documents, and correspond with legal compliance analysts through the same interface.

2.2.3 Returns

At each provider, process requests are typically managed by a combination of paralegal staff and lawyers. They will review the document for facial consistency (i.e., that it is well-formed, properly signed, addressed and dated, provides the appropriate level of process for the data requested, etc.), validate that they are capable of obtaining the data requested, and determine whether or not there are any policy objections with legal recourse (e.g., a determination that a subpoena's request is overbroad). It is our understanding from our interviews that such reviews typically take a modest, but single-digit, number of minutes to complete. If an order passes review, the organization executes the data searches associated with the identifiers (i.e., accounts, phone numbers, etc.) and requests (i.e., email, videos watched, etc.) specified in the order.

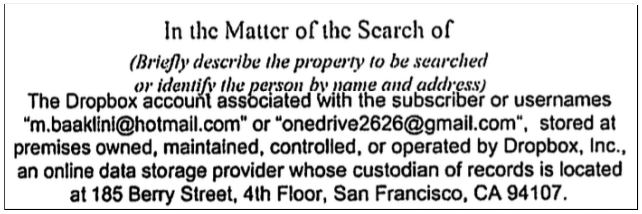
When the resulting data is assembled—commonly called a "return"—it is then provided to the requesting party. In portal systems, returns are typically provided through the same authenticated online system, and agents will be notified to download their returns directly from the portal. Otherwise, it is common practice for the request to include a direction to transmit returns to the requesting officer at a particular email address included in the request. Returns may also be sent via postal mail or courier. Finally, for many forms of process—including warrants—agents are required to file notice with the court that they have received returns for the court's order.

3 Threat Model

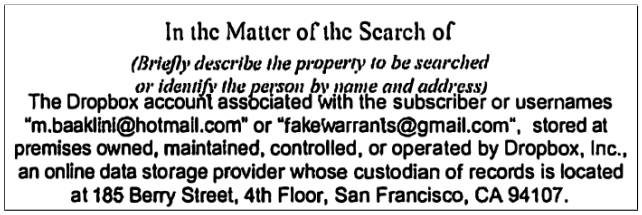
Our threat model focuses on adversaries with the capability to forge unauthorized process documents (e.g., search warrants, subpoenas, and 2703(d) orders), and the ability to convincingly serve them on a provider (e.g., via fax, e-mail, or a compromised portal account) but who are *not* able to obtain original signed process documents of their choosing. This model captures virtually all known existing attacks on process integrity (including forgery or modification of existing process by law enforcement agents), but it does not include attacks that subvert the underlying authorities in the legal

[5] This detail is corroborated by our interviews with law enforcement and court staff, and also is clearly evident in the structure of modern criminal process documents found in PACER. We note that court staff, while highly skilled professionals, generally are not technologists. Thus, there is tremendous variability in how documents are manipulated (e.g., whether PDFs are flattened, how signatures or other markings are applied, scanner settings, etc.) and in the understanding of the security implications of those choices.

[6] Tracking these individualized procedures can be complex; both nonprofit (i.e., The National Consortium for Justice Information and Statistics, from which our examples were gleaned [48]) and for-profit entities (e.g., `warrantbuilder.com`) offer services to help navigate these requirements.



(a) An unmodified warrant



(b) Substituting an identifier with `fakewarrants@gmail.com`

Figure 1: Search warrant modification with Adobe Acrobat.

system itself—such as law enforcement agents who submit fraudulent requests to judges targeting innocent parties of their choosing [38], nor judges, or court clerks who purposely misuse their authority to empower illegitimate process. We briefly discuss our two capability assumptions here: document forgery and document service.

Forging Process Documents. Our experience is that forging legitimate-looking government legal process does not require sophisticated technical skills. One basic approach is to 1) find a valid example document (e.g., via PACER, the public-access repository of CM/ECF documents), 2) modify it using off-the-shelf tools to target a different individual in lieu of the original target, and 3) obfuscate that the document was modified. For example, Figure 1 shows an excerpt from a seamless modification to a Federal search warrant using only Adobe Acrobat Pro. Notably, with minor effort the new text can even replicate noise in the document (see Figure 1b, the modified text has some distortions that match the rest of the paragraph). After a PDF is modified, tools such as `exiftool` [23] can adjust the metadata to obscure changes and appropriately update the creation date.

Serving Forged Documents. As we have described, there is no inherent authentication for process service (i.e., providing evidence that it is actually being served on behalf of a legitimate law enforcement agency). Thus an attacker's required capabilities are relatively modest, including any of: 1) the ability to create a fake account on the portal that a service provider operates for receiving legal process; 2) access to a compromised account on the portal; 3) access to a compromised email account of a law enforcement agent; or 4) ability to fax or mail forged process to the provider that accepts that form of service. In many of the public cases documenting forged legal process, hackers compromise government email addresses (either via compromising the agency website, or through phishing or password guessing) [37] and these email addresses are then used for process service. We also consider a more capable adversary, e.g., a rogue law enforcement agent, who modifies an existing signed process document and then sends it to the provider [27].

4 System Requirements

Given this threat model, our goal is to allow the recipient of a criminal process document to validate that it was duly authorized (i.e., that it was, in fact, signed by someone with the legal authority to issue such a demand) and that its contents have not been meaningfully altered since authorization. However, in our discussions with judges, court clerks, law enforcement, and providers, we have identified a set of real-world constraints that impose concrete engineering requirements on any practical design. We describe each in turn here:

Self-Certification. In principle, it is appealing to imagine that service providers could simply contact an issuing courthouse to validate the integrity of an order on its authority. However, criminal process is typically sealed—meaning that it will not appear on any public docket and court clerks are not allowed to respond to queries about it (even if they had the staff to do so). This practical reality creates an implicit requirement that legal process documents be self-certifying.

Paper Compatibility. The traditional way to accomplish self-certification is via digital signatures computed over the entirety of a document (or a one-way hash thereof) that are embedded as metadata. Any recipient could then use such a signature, along with a public key associated with the authorizing party, to verify that a signed document was issued by the authorizing party and unchanged since it was signed.

Unfortunately, end-to-end digital document processing is not feasible in the existing U.S. legal system. Paper documents and/or scanning are still an important and commonly-used part of the process, both in process creation and in service—thus, there may be no end-to-end channel for holding digital metadata like a signature. Moreover, any "whole document" digital signature is too fragile to accommodate the common practice of minor edits or transformations to process documents (e.g., even after they are authorized via a judge's signature). These facts create two further implicit requirements: 1) any verification data must survive in the physical domain (i.e., after being printed, faxed, scanned, folded, etc.), and, 2) any verification method must accommodate incidental textual changes to the document that do not impact its semantic meaning (i.e., what data it targets, what kind of searches it authorizes, and the authority upon which it draws). Thus, we believe that a simple "verified vs. non-verified" signal is

unlikely to be achievable, and any practical scheme will need to provide more fine-grained information to analysts (e.g., which part of a document may have changed).

Ease of Use. From our interviews, it is clear that while judges and court staff are highly skilled individuals, they are not technologists. Solutions that require them to correctly make specific technical choices are unlikely to succeed (e.g., we found at least a half-dozen different ad hoc mechanisms used to apply judicial signatures to PDF documents). Similarly, law enforcement agents can also be imperfect stewards for the process documents they serve (e.g., we have found situations where Federal agents have filed screenshots of process documents displayed in their browser—including all of their open browser tabs—in lieu of the documents themselves; and this works!). Finally, corporate legal analysts (i.e., those reviewing received process documents) are also not necessarily tech savvy, and, in evaluating documents for potential integrity issues, they require signals that align with their own understanding and expertise. Taken together, these create a requirement for a self-certification process that is simple to apply (e.g., automatically incorporated into an existing workflow, such as document filing), difficult to inadvertently perturb (i.e., survives printing, faxing, minor modifications, etc.), and is straightforward (for a paralegal) to interpret.

Incrementality. Finally, there is considerable diversity and independence in the Federal courts. Each district operates as a fiefdom unto itself and, thus, new technology—and policies around its use—are rolled out at different times in different jurisdictions, with different levels of support. Indeed, even upgrades between versions of the national CM/ECF system are managed independently by each court. Therefore, any solution will need to interoperate with existing process document practice for quite some time.

5 *WarrInt*

We have designed a prototype system, *WarrInt*, to align with these requirements. Our design goal is to dramatically improve the ability to identify process forgeries, while still being practically deployable in the U.S. legal ecosystem. In this section, we share our design approach, describe how such a system might integrate into existing court and provider workflows, and provide implementation details of our prototype.

5.1 System Design

Our solution to the practical constraints we have documented is based on two key design elements:

- *Paper-based digital channels*. Since there is no reliable digital channel for transmitting verification data (including any digital signatures) we must embed such metadata directly into the printed form of the document itself—treating the paper as a channel for encoding digital data. Thus, our system modifies each process document to add one page of two-dimensional matrix barcodes (henceforth, just barcodes) to encode its verification data.
- *Semantic integrity validation*. Since documents may be benignly modified after authorization (e.g., via adding case numbers, stamps, seals, CM/ECF headers) or may change their binary representation during service (e.g., due to rescanning, faxing, etc.) a traditional binary integrity check over the whole document has limited utility. Instead, we focus on the integrity of the document's semantics: *is the search demanded by the document consistent with the search authorized by the judge's signature?* Specifically, the system extracts the words and their placement present in the document and encodes those into the verification data (i.e., the barcodes). Then, when a recipient views the document served upon them, they can easily compare the document with the information encoded in the barcodes to see what specific locations in the document have changed after signing.

More concretely, our basic design approach is as follows: after a process document is authorized (e.g., signed by a judge), the system extracts the content from each source page and encodes it into a digital representation, which we call the *document description*. We have chosen to use optical character recognition (OCR) for this purpose for two reasons. First, legal process is almost entirely text-based and thus a document's words and their placement on the page implicitly encode the document's semantics. Second, OCR is naturally robust to transformations in the analog domain (e.g., the distortions that arise from scanning, faxing, printing, etc.).[7]

The system then signs the document description using the private key of the party authorized to issue the process (e.g., the judge or court). The resulting signed document description, along with a certificate attesting to the public key's validity, is encoded as a printable 2D-barcode similar to a dense QR code, and appended to the document. Due to the digital signature, an attacker cannot generate a legitimate barcode page without compromising a judge's or court's private key.

When such a document is served upon a provider, the system decodes the barcode at the end of the document which reveals the words and their coordinates from the original document. If the certificate and signature verify, then this embedded document description is presumed to be valid. Next, the semantic feature extraction process is repeated on the *body* of the document, using OCR to extract a new, local version

[7] We briefly explored image-domain fuzzy hashing approaches (e.g., the PDQ perceptual hash algorithm developed by Meta [16], or NeuralHash developed by Apple [6]) but found that they were substantially less sensitive to semantically relevant changes, and thus allowed far more false negatives (i.e., undetected forgeries).

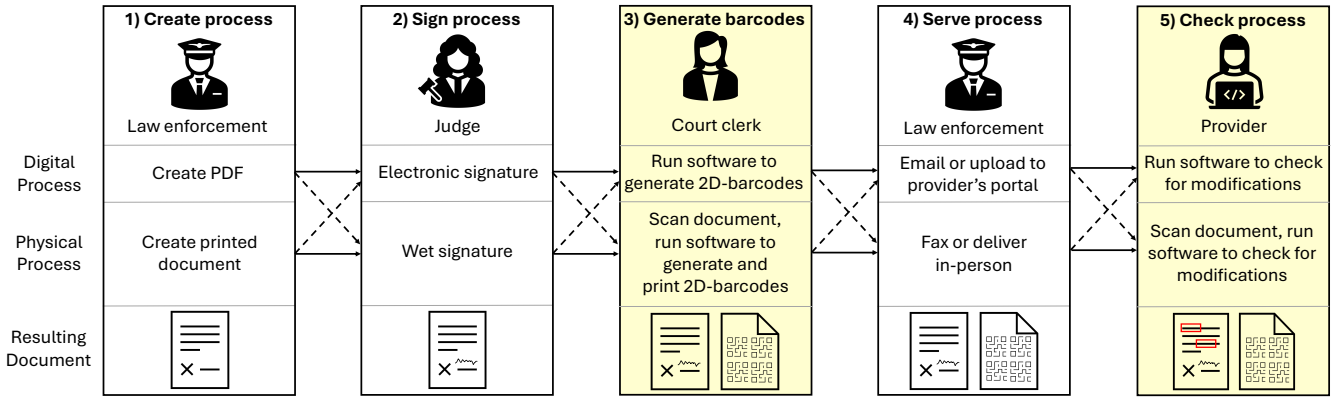


Figure 2: Overview of *WarrInt*. Solid lines indicate continuing in the same modality (digital or physical), and dashed lines indicate switching modalities (through printing or scanning). Highlighted boxes indicate where *WarrInt* is run.

of the document description. Finally, this local document description and the signed document description decoded from the barcodes are compared to identify any inconsistencies.

We have specifically chosen not to automate the interpretation of this comparison. The semantic importance of individual parts of a process document can depend on a range of facts, including specific information about the target service (e.g., the format for valid identifiers), the kind of process (e.g., warrants have explicit periods of validity while subpoenas do not), as well as interpreting different form versions and ways to structure requests in English. Rather than attempt to encode that knowledge into a tool, we instead visualize the differences for a legal analyst to evaluate, based on their own domain knowledge. Thus, our tool highlights places in the document that are inconsistent (i.e., there is a difference between the original document description and the one extracted from the document they see before them) and can interactively compare the two versions. An analyst is best situated to determine if the location and content of an inconsistency is semantically relevant (e.g., changes to the email to be searched)—which we call *salient inconsistencies*—or merely a byproduct of noise (e.g., from court process vagaries, scanning, or our encoding/decoding process).

5.2 Deployment Sketch

Figure 2 outlines how *WarrInt* could integrate within the existing criminal process creation workflow. It is common practice that court clerks manage such documents, scanning them if they were physical (i.e., received a "wet" signature), adding the case number and any seals (per the practice of the courtroom and district), filing them into the court's record systems, and making the result available to the requesting Assistant US Attorney and/or law enforcement agent. The centrality of the clerk's role here makes them an ideal implementation point for our software. Thus, we envision that *WarrInt* is transparently integrated into the clerk's filing process, wherein our

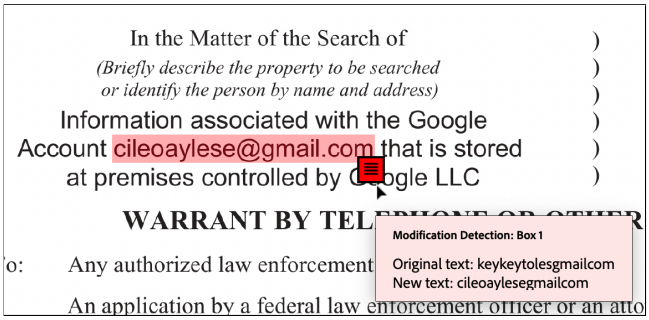


Figure 3: Example of how our system flags modifications to an analyst. The analyst need only hover their mouse over the red box, and a popup displays the original and changed text.

software would use OCR to capture the document description, digitally sign it with the most granular key available,[8] encode the digital stream as barcodes and append this data to the source document. It is this document that would be produced to law enforcement for service.

As we have described, the document may then undergo changes (including additional scanning and transformation) as a part of law enforcement handling and service before finally arriving at the provider. The provider side of *WarrInt* processes the received documents to validate the certificate and signature, and then compares the calculated document description (i.e., via local OCR) with the encoded and signed document description in the barcodes. *WarrInt* then highlights any region of the document that differs between the two, to allow legal analysts to evaluate if it is problematic (shown in Figure 3). If so, the provider can reject, request a new copy from law enforcement, or perform any other form of out-of-band process to "cure" the inconsistency.[9]

[8] Clerks typically work for a particular judge or magistrate, but they also already specify the presiding judge as part of the filing process, so such a key selection could easily be automatic.

[9] In discussions with providers, we learned that they may contact the issuing agent and/or the responsible AUSA out-of-band for further clarification.

We note that our use of digital signatures for integrity presupposes the existence of a PKI for the Federal courts. The Department of Justice already uses the existing Federal PKI [62] to issue each Assistant US Attorney a unique Personal Identity Verification (PIV) card containing a certificate and key pair. But a counterpart for the Federal courts would need to be implemented. Ideally, courts would issue certificates for each individual district judge and magistrate. We can also imagine a stopgap implementation where a single public key would represent a district—mirroring existing practices where each district court obtains a unique TLS certificate for its website.

5.3 Prototype Implementation

The *WarrInt* prototype is implemented in Python with two components: one for process issuers and the other for receivers, closely following the design we have described.[10] The system works exclusively with PDF files—the issuing component takes a PDF as input and produces a modified PDF with an appended, signed, document description in barcode form. The receiving component takes in such a PDF as input and produces a new PDF, annotated with boxes indicating any potentially modified regions of the page, with their detected change (e.g., Figure 3; see Appendix E for examples of search warrants, administrative subpoenas, and 2703(d) orders). This PDF-centric approach ensures that *WarrInt* is compatible with existing storage, transport, and processing systems and, on the provider side, that investments in existing analyst client software is unaffected.

In the remainder of this section, we focus on the implementation details involved in extracting, encoding, and comparing the document descriptions, which represent the most significant technical challenges in *WarrInt*'s implementation.

5.3.1 Extracting Document Description

Our prototype implementation uses Google Document AI's "Document OCR" processor.[11] We use the OCR tool to extract *tokens*, which conceptually map to words, but may also include punctuation (e.g., both "-" and "1-1" may be tokens). For each token, we record the text and the coordinates of the corresponding bounding box. These tokens and their bounding boxes make up the document description that we sign and encode using barcodes affixed to the process document. We disable the OCR tool's use of any digital contents (i.e., for non-scanned documents with embedded text), as our experience is that it degrades overall performance in practice due to stamps, seals, and other non-salient graphical features as discussed below.

[10] One exception is that the prototype does not yet implement a full cryptographic certificate and simply embeds the public key of the signer.

[11] In particular, version `pretrained-ocr-v2.1-2024-08-07`. We tested other tools like Tesseract [55], Amazon Textract [4], and Adobe OCR [3], but Google's Document AI performed most reliably for this application.

5.3.2 Paper-Channel Encoding

WarrInt encodes the document description and a digital signature[12] over that description as a barcode so that the provider can verify the document regardless of whether it is served digitally or physically. To minimize the burden on courts, we seek to encode this digital information in a single appended extra page. Before transmitting the document description, we encode it as follows: tokens are encoded directly (each token is prefixed with a byte containing the token's length), then we encode the bounding boxes as two placement coordinates (left and top) and two size values (width and height). We quantize the width coordinate into a single byte (we conduct our analysis at 300 DPI), and group coordinates by type (i.e., all the left coordinates together in order). Finally we compress this stream with Lempel–Ziv–Markov chain algorithm (LZMA).

We encode the resulting data into the paper channel using the Twibright Optar system [57], which, akin to QR codes, encodes binary data in a 2D-matrix barcode format with error correction via Golay [29] codes. By default, Optar delivers a carrying capacity of 200 KB per page but requires scanning at 600 DPI; we adjust Optar's settings to target 200 DPI (since we find that 200 and 300 DPI are common scanning resolutions for real legal process, c.f. Section 6) resulting in a capacity of 16.3 KiB/page, which is sufficient for our needs.

5.3.3 Receiver-side Scan Registration

One challenge occurs because documents may be rescanned *after* the original document description is calculated (e.g., due to fax or other service vagaries). Scanned documents can depart from the original in systematic ways, e.g., the scanned document may be rotated or magnified. We find such page-wise transformations can significantly impact the number of inconsistencies that *WarrInt* highlights, so we regularize the process documents at the service provider. While there exist computer vision techniques to project one copy of a document page (i.e., the process document received by the provider) into the plane of another (the original issued by the court), these approaches generally use image-based features (e.g., detected edges of page content) to compute a *homography matrix* that mathematically transforms coordinates collected from one copy into corresponding locations on the other. Unfortunately, this is impractical for our use case, as we need to communicate the features across the limited-capacity paper channel—in barcodes—and image-based features are quite numerous.

Instead, we develop a technique described in Algorithm 1 and detailed in Appendix D that relies exclusively on the information contained in the document description. In short, in contrast to traditional image-based pipelines that rely on local gradient descriptors (e.g., SIFT [41]), we treat OCR tokens as high-level geometric features, each represented by their spatial centroid and textual content. This provides enough in-

[12] Our prototype uses Ed25519 keys for EdDSA signatures.

Algorithm 1 Homography Estimation from OCR Features

Require: OCR outputs from source and target documents: $\mathcal{W}_s, \mathcal{W}_t$
Each $\mathcal{W} = \{(x_i, y_i, w_i)\}$: word centroid and string
Ensure: Estimated homography matrix $H \in \mathbb{R}^{3\times3}$
Filter stopwords and whitespace from $\mathcal{W}_s$, $\mathcal{W}_t$
Build bipartite graph on word string similarity: $G = (\mathcal{W}_s, \mathcal{W}_t, E)$
Solve max-weight bipartite matching to find word correspondences $\mathcal{C}$
for all duplicate matches **do**
    Select minimal pixel-distance correspondence
end for
Apply RANSAC on $\mathcal{C}$ to estimate homography matrix H
return H

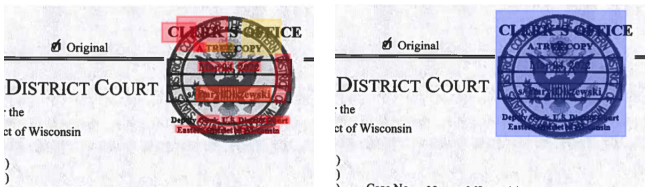
formation to form correspondences that reveal the geometric transform between documents, necessary for alignment and registration of a candidate document to an input template.

5.3.4 Line Formation

Unfortunately, even perfect page registration does not guarantee identical OCR results. While Google Document AI empirically produces deterministic results for a fixed input PDF, minor differences induced by scanning (even at a high resolution) can produce different outputs. In particular, we note that the OCR tool may split up words into tokens inconsistently (e.g., an email address, `fake.warrants@example.com`, may be parsed as one token at the court, but as two tokens, `fake.` and `warrants@example.com`, when parsed after scanning at the provider). Moreover, even if the OCR tool extracts the same set of tokens, they may output with slightly different bounding-box coordinates due to quantization effects in the PDF rendering, homography matrix calculation, or both. As a result, it is not always trivial to determine the correspondences between tokens across the two copies of the document.

In order to construct a robust correspondence, we leverage the fact that process documents are generally single columns of text, with limited free-floating text. Hence, we group tokens into lines by sorting them by the top of their bounding box and grouping together tokens whose bounding boxes overlap vertically by at least 60% of the median height of the bounding boxes extracted from the document—which accounts for varying font sizes used in process from different courts.[13] We then concatenate the tokens in each line into one single text string ignoring whitespace and punctuation. This approach works well in practice, but it is valuable to note that should it "fail," this will lead to *more* differences being (spuriously) flagged and will not cause any real inconsistency to go undetected.

[13] Empirically, this 60% threshold is optimal for our training dataset. We also tried using Google Document AI's built-in line grouping feature, but our approach provides superior performance empirically.



(a) *WarrInt* before seal detection. (b) *WarrInt* with seal detection.

Figure 4: A document where automated seal detection eliminates two-dozen spurious inconsistencies around the seal.

5.3.5 Seal Detection

One issue with line-by-line processing is the fact that many process documents are decorated with seals and/or stamps added at the courthouse, which themselves contain text that is both unanchored to the rest of the document and rendered at an angle or along the curve of the seal itself. Even if the OCR tool is able to extract the text from within the seal (which depends greatly on the particular seal), it is not obvious with which line of the document to associate it. As a result, it is challenging to compute robust correspondences between such text in the original and served documents.

We address this issue by observing that a stamp or seal—and any text it may obscure—is highly unlikely to contain information salient to the document's validity (i.e., it should not contain or occlude any identifiers of the target of the process).[14] Hence, we developed an AI-based heuristic to detect stamps and seals and their bounding boxes in the document received by the provider [34].

Service-provider analysts can optionally enable *seal filtering*, which excludes from consideration OCR tokens that overlap with seal bounding boxes. The filtered area is highlighted in blue, as shown in Figure 4, allowing the analyst to confirm that the exclusions are appropriate or, if necessary, disable the filter. As shown in Section 7, seal filtering dramatically increases the usability of our tool for process documents that contain graphical elements. We emphasize that this filtering is performed exclusively at the provider side; the document description affixed to the document contains all of the OCR tokens extracted at the courthouse so attempts to evade detection by deliberately adding a seal before service will not prevent an analyst from viewing any inconsistencies.

5.3.6 Highlighting Inconsistencies

Finally, once we have grouped tokens into lines, we compare the lines in both copies of the process document (the original document at the time of the judge signing, and the document at the time the provider receives it) to identify inconsistencies, defined as an insertion, deletion, or replacement of any char-

[14] Such seals and stamps are almost always either in the header of the page (typically where the issuing court is identified), or at the bottom of the page around the judge's signature. We have never seen such a decoration that overlaps with salient information.

acter in these lines. Our tool visually pinpoints the location of these inconsistencies by drawing colored boxes over them and providing an annotation that includes the corresponding original text (Figure 3).

6 Process Document Dataset

Both the design of our system and its evaluation is closely driven by the nature of existing Federal criminal process documents. Unfortunately, there is no extant corpus of such documents. Indeed, as we have mentioned, the vast majority of criminal process is sealed by the court, and providers are loathe to share documents that implicate their customers' privacy interests. Thus, we have created our own corpus by manually searching for Federal criminal process documents that are unsealed.[15] To this end, we have searched across PACER (the public version of CM/ECF) for cases docketed in the last three years that contain legal process targeting online service providers (i.e., search warrants, 2703(d) orders, and a small number of administrative subpoenas with any attached 2705 preclusion orders).[16] For each such docket, we used the court's CM/ECF portal to download process applications, orders, and/or returns that were available as PDFs (sometimes a docket may not be sealed, but individual documents are).

Ideally, the court's files would provide us with complete documents in precisely the form provided to law enforcement and then served on service providers—what we call a *canonical document*. Unfortunately, this is not always the case. For example, in some cases we may obtain a search warrant application, but not the issued warrant itself, or an issued warrant without attachments, or a copy of a warrant, filed in a return, and so on. In such instances, we synthesize a document as close to canonical form as possible using some combination of the issued order, the application for an order, attachments to the application, and copies of the order as filed in a return. For example, one might compose the attachments from a warrant application with the warrant order page as extracted from a filed return. In each case, we attempt to 1) use the document closest to the original process document (e.g., we prefer returns that include the original warrants to warrant applications), and 2) capture such documents in the modality (scanned vs digital) that we believe the original process was recorded. Of the 257 total documents we collect, two were already in canonical form, and 159 only required removing extraneous pages. The remaining 96 documents required synthesizing multiple documents.

[15] This can occur because the original filing was not sealed (e.g., because the target of the search was already in custody), because certain districts automatically unseal documents without a specific motion to keep them sealed, or due to litigation, among other reasons.

[16] There is no explicit mechanism to query for such documents, so as a proxy we have accomplished this by querying for case names such as `"USA v. *.com"`, which reflects the most common idiom for such dockets, as well as by searching for docket entries identifying "search warrant" or "2703", along with provider recipients, and then validating by hand.

While we are unable to make any claim of representativeness (since there is no ground truth dataset to compare against), we have taken care to curate a diverse dataset from across many courts and across our time period. Our dataset's 127 canonical process documents span 25 district courts, covering all but one of the Federal circuits, and these districts account for over 60% of issued search warrants during the time period [59], suggesting that they are unlikely to be atypical. There is also some variation in source modality. Of our 127 documents, 34 are scanned, but the vast majority have a combination of scanned and digital content.[17] For the scanned documents, we use the `pdfimages` tool to identify the source DPI and find that 300 DPI dominates with outliers ranging from 150–400 DPI.

We refer to this dataset as our *training set*. Although, aside from the seal detector, our system does not require "training" per se, we are sensitive to the fact that we have designed our system in the presence of this data. Thus, in the evaluation section we introduce a separate test dataset of 130 documents (created similarly, but drawn primarily from distinct district courts and, unlike the training dataset, containing 2703(d) orders) to ensure that our results do not arise from implicit selection bias.

7 Evaluation

In this section we present an evaluation of *WarrInt*. At a high level, we simulate process service by running each of the canonical documents through *WarrInt*: we replicate the courthouse by inputting the canonical document as retrieved from PACER to generate verification barcodes and emulate verification at the service provider by providing the process document—and barcodes—in different formats. We conduct two separate sets of experiments intended to quantify security and utility: in the first set, we alter the identifiers contained within the process documents after generating the verification barcodes to evaluate how effectively *WarrInt* thwarts attacks (i.e., *how well does the system work?*). In the second set, we make no deliberate modifications and assess the extent to which *WarrInt* generates spurious inconsistencies (said another way, *how much effort is required by the analyst?*).

7.1 Methodology

We replicate "digital" service (e.g., email) by providing the PDF with barcodes directly to the service provider; we replicate the "paper channel" by printing and scanning the document at various resolutions using a high-volume Xerox scanner in heavy use by an educational institution in an attempt to mimic conditions at a busy courthouse. CM/ECF

[17] To identify scanned documents, we compute the *area* occupied by bitmap data and for printed glyphs. Empirically, when this "text-to-image" ratio is below 0.1, the documents were clearly scanned.

	Min	Median	Mean	Max
PDF				
Digital	8.82	135.20	86.03	1044.57
200-DPI scan	14.59	66.90	71.84	139.17
300-DPI scan	19.29	130.35	135.67	272.64
400-DPI scan	28.45	227.86	252.23	453.35
OCR				
Raw OCR output	22.34	39.57	39.27	73.45
Document descr.	2.06	3.60	3.59	6.82
WarrInt	0.44	1.51	1.50	3.91

Table 1: For the 127 documents in our training set, we report the size in kilobytes (KiB) per page of the digital PDFs (with fonts embedded), scans at various DPIs, the corresponding digital PDF's OCR output, the encoded document description, and our compressed binary representation thereof (*WarrInt*).

automatically applies a digital header to all documents uploaded to PACER which is not usually present on legal process that providers receive. In our experience, these headers are frequently outside the area of the page that scanners consider, and, even if they are included, often occlude other (non-salient) header text. In order to avoid impacting our results with these artifacts of our dataset, we exclude the top 5% of each page from *WarrInt*'s analysis.[18] We indicate this exclusion to the analyst in the same fashion as with seal detection: by placing a blue ribbon over the ignored area.

To ensure that *WarrInt* is not overly tuned to the training dataset we used during development, we employ the same methodology as described in Section 6 to assemble a separate *test dataset* of 130 different documents that we use exclusively for evaluation. Like the training dataset, the test dataset includes search warrants and administrative subpoenas, but also includes 2703(d) orders, which have a different format.

7.2 Efficiency

Before considering the effectiveness of *WarrInt*, we first quantify its overhead in terms of the amount of additional data (i.e., the size of the document description) that needs to be affixed to process documents as a set of barcodes for each of our canonical documents. As shown in Table 1, *WarrInt*'s document descriptions are significantly smaller than than the PDFs themselves, and our binary encoding scheme reduces their size even further, achieving one-to-two orders-of-magnitude reduction compared to the document itself. While some pages are more information-dense than others, we find that across all legal process documents in both datasets—which range from 2–16 pages in length—we need only one page of barcodes to store the encoded information (i.e., the total size of the encoded document description is less than 16.3 KiB).

[18]This threshold was empirically determined using our training dataset to cover the CM/ECF header in most cases without including any salient text.

In comparison to Optar, QR codes would require significantly more pages. To reliably recover information in QR codes scanned at 200 DPI, we empirically find that we needed to use Version 20 (V20) QR codes (which can store 276 bytes of payload each). After allowing the recommended 15% quiet zone around QR codes for maximum rate of detection [14], we can store 12 QR codes per U.S. Letter page, with a total carrying capacity of 3.2 KiB per page. In this setup, our largest legal process document (with a 15.9-KiB document description) would require five pages of QR codes—compared to only one page with Optar.

7.3 Attack Success Rate

This set of experiments measures how well *WarrInt* detects adversarial modifications. For each piece of legal process in the training and test datasets, we select one identifier to modify. We perform two experiments: one replaces the entire selected identifier with a randomly generated string, and one replaces one character of the identifier to emulate an attack that tries to exploit common OCR mistakes, acting as a worst case for our system. We enable seal detection in all cases.

7.3.1 Identifier Modification

We seek to evaluate *WarrInt*'s security against an optimal attacker who makes the minimal changes necessary to the legal process document. We hypothesize that the best way to launch such an attack would be to modify a digital copy of the process document—i.e., the PDF itself, before it is rendered—so that the modified identifier is rendered in precisely the same font and without introducing any unwanted artifacts. We emulate this attack by converting the process PDFs to JSON using the `cpdf` command line utility [12], modifying the JSON to alter the identifiers, and converting the JSON back into a PDF. We confirm that the resulting PDF differs only in the desired way.[19] This approach requires us to restrict our focus to documents where the identifiers are rendered using digital text—as opposed to bitmap images, which sometimes appear in even non-scanned documents.

Completely Replacing Identifiers. In our first experiment, we emulate an "average case" attack by replacing an identifier in the process document with a randomly generated string. We generate random replacement identifiers while ensuring that the replacement is not so wide that it will overlap with the subsequent text in the PDF. To achieve this, we calculate the width of the rendered identifiers and check if a randomly generated replacement of the same length as the identifier has a width less than 95% of the original identifier. We attempt 100 randomly generated identifiers, and if none achieve this width constraint, we decrement the number of characters in

[19]In contrast, we find that modifying documents with standard PDF editors like Adobe Acrobat result in many unintentional changes to the document.

Format	Training Dataset					Test Dataset				
	Min	Mean	Median	p90	Max	Min	Mean	Median	p90	Max
Digital	0	0.02(0.03)	0	0	8(9)	0	0.02	0	0	14
200 DPI	0	2.22(3.14)	1	6(9)	51(56)	0	2.24(7.09)	0(1)	6(20)	69(70)
300 DPI	0	1.90(2.67)	0(1)	5(8)	44(61)			–		
400 DPI	0	1.85(2.63)	0(1)	5(8)	48(74)			–		

Table 2: The number of spurious detections per page. (The numbers in parentheses are without seal detection.) Due to the manual nature of printing and scanning, we only report results for the test dataset scanned at 200 DPI.

the replacement identifier and continue to generate candidate replacements. We create a numeric replacement identifier if the original identifier is numeric, otherwise alphanumeric. We successfully modify 81 of the 127 documents in the training dataset and 102 of the 130 documents in the test dataset.

Leveraging OCR Confusion. In our second experiment, we consider an attacker who is able to find an existing process document that contains an identifier that is extremely close to their desired target: namely, one whose difference consists of exactly one character that is commonly confused by OCR (e.g., 'l' instead of 'i'). We rely on the character confusions documented by Jatowt *et al.* [35]. In particular, we compile the single-character confusions documented for two datasets comprised of modern news articles (Overproof NLA and Overproof NC). Because these datasets only report character confusions for lowercase characters, we only modify documents that contain identifiers with lowercase characters. We use the percentage of each character confusion in the datasets (choosing the larger percentage if a confusion occurs in both Overproof datasets) as weights when we randomly choose a character to replace in the identifier. For example, the character 'e' is confused as an 'o' in 14.8% of occurrences in the Overproof LC dataset, and, thus, is replaced in a similar fraction of the modified documents. We create 63 modified documents from the 127 documents in the training dataset, and 82 documents from the 130 documents in the test dataset.

7.3.2 Results

Because all of the modified documents are digital, we can consider both electronic (e.g., emailed PDF) and paper (i.e., scanned) service modalities. Hence, we evaluate *WarrInt*'s performance on the modified PDF, as well as with a printed-and-scanned version. We scan the modified documents at 200 DPI to consider the worst-case scenario, as OCR performance improves with higher DPI. We measure the attack success rate per modification and per document (i.e., evading all modifications in a given document).

Overall, *WarrInt* is effective at detecting modified identifiers. We find that *WarrInt* catches all modifications (i.e., ensures an attack success rate of zero) for both digital and physical service on the training dataset. On the test dataset, *WarrInt* catches all modifications when identifiers are replaced with random strings for both digital and physical service; however, it misses one OCR error, failing to flag one of two instances in the same document where "agent" was changed to "agcnt," when tested on both the digital and scanned versions. *WarrInt* misses another OCR-confusion modification only when the documents are scanned: *WarrInt* does not detect an 'i' being replaced with an 'l' in one of two instances of the identifier. Overall, these three missed detections result in a per-modification attack success rate of 3 out of 640 modified identifiers across all documents and modalities, or 0.5%. However, for both of these documents the modified identifier appeared at least twice in the document and *WarrInt* correctly flagged at least one instance of the identifier as modified, which would give a careful analyst pause. Hence, we argue that *WarrInt* provides an effective document-wide attack success rate of zero in all cases.

7.4 Spurious Inconsistencies

We also study the rate at which *WarrInt* "detects" an inconsistency that turns out to be spurious (i.e., how often it flags portions of a document as containing alterations when in reality there are none) as well as the potential saliency of these purported inconsistencies (whether an analyst is likely to be concerned by the highlighted region). These are measures of usability as analysts will quickly tire of a system that generates too many false alarms that cannot be quickly dismissed.[20] While the frequency of spurious detections impacts the level of effort required by the analyst to validate a process document, their saliency speaks directly to the issue of whether an analyst is likely to make the wrong decision, i.e., reject a valid process document. In general, the extent to which *WarrInt* will produce spurious alerts depends on the perturbations documents go through during service. Here, we consider the base cases, i.e., where the document is served via either a digital or paper channel, but leave an evaluation of mutilation due to handling (e.g., folding, crumpling, coffee stains, etc.) and how that may facilitate attacks to future work.

[20] Again, for context, we understand that existing analyst review typically involves a modest number of minutes in practice.

7.4.1 Frequency of Spurious Detections

Once again we consider both electronic and paper delivery channels, analyzing both the actual PDF and versions printed and scanned at three different resolutions—200, 300, and 400 DPI—representative of the resolutions we encountered in PACER. Table 2 reports the number of spurious inconsistencies reported per page by *WarrInt* across the 1,416 pages of both the (710-page) training and (706-page) test datasets in the absence of deliberate modifications. (The prevalence of spurious inconsistencies in the altered documents discussed in Section 7.3 is similar, so not reported separately here.)

Overall, the results are consistent across resolutions and datasets, with 90% of pages having six or fewer spurious alerts, with the median page having at most one detection even when scanned at 200 DPI. The average page has approximately two spurious detections, although this is skewed by a few documents that contain pages for which *WarrInt* produces a large number of spurious detections regardless of scanning resolution. (We note that while the digital channel is not entirely free of spurious alerts—due to quantization effects—they are rare.) Seal detection removes one spurious inconsistency on the average page in the training dataset and has an even greater impact in the test dataset: approximately five fewer spurious detections in the average case.

Manual inspection of the ten pages with the highest number of spurious inconsistencies (six from the training dataset and four in the test set) reveals that 90% (375 out of the 415 total detections across these 10 pages) of the spurious detections are due to incorrect line groupings (an example of which is shown in Figure 5 in Appendix E). OCR confusion gives rise to 2.5% of the inconsistencies and another 2.5% occur in one specific document where the PACER version was a low-quality scan. The remaining 5% are a hodgepodge of handwritten content and signatures, undetected stamps, and CM/ECF headers outside of our filtering region.

7.4.2 Saliency of Spurious Detections

Many detections can quickly be dispensed with by an analyst because they highlight innocuous portions of the document (such as pro-forma text or even scanning artifacts like dust specks). The real burden falls on analysts who are forced to consider inconsistencies involving identifiers. For a random sample of 100 documents scanned at 200 DPI (58 from the training dataset and 42 from the test dataset) with at least one spurious inconsistency, we manually check if the inconsistency involves an identifier, which may cause analysts to scrutinize—or even reject—a valid process document. We find this occurs 20% of the time in our sample (16 documents in the training dataset and only four from the test dataset), and almost half the time—eight instances—the discrepancy arises because the OCR tool mis-parses the '@' symbol in an email address as the character 'a'. Of the 12 documents where an inconsistency is flagged within salient part of an identifier (i.e., the user or domain name), all but one are due to OCR confusion, typically involving the characters 'l', 'i', and '1'.

8 Related Work

To the best of our knowledge, our work is the first to consider the (lack of) integrity guarantees for criminal legal process. However, we discuss three classes of related research that tackle associated issues of legal accountability, forgery detection, and document authentication.

Cryptographic Legal Attestations. There have been several efforts to use cryptography to provide *new* capabilities in court documents. For example, Frankle *et al.* develop a system using multi-party computation to produce aggregate statistics about sealed legal documents while maintaining their required secrecy [28]. Similarly, Kroll *et al.* apply identity-based encryption to design a system to improve the secrecy of court orders—for example, even hiding the subject of the order from the provider itself [39]. However, while both systems are adjacent to the problem we consider, neither are motivated by the need to protect process integrity. Moreover, both are decidedly "blue sky" designs, while our work is very much driven by the challenges of practical integration into the existing legal ecosystem.

Document Tampering Detection. Another related line of work focuses on detecting general document tampering in hardcopy (i.e., non-digital) documents. Most of this work implicitly models the would-be forger as deviating from inferred local image properties. For example, Diarra *et al.* [17] devise an unsupervised approach which focuses on small segments of text or images using self-attention, and identifying "intrinsic" features of the document's visual and textual content. It then flags any deviations from these qualities. Similarly, Gorai *et al.* [32] propose an approach (tested on handwritten documents) to automatically model features of the ink used in a document to detect local deviations.

Document Authentication. Yet other approaches seek to authenticate documents via the addition of some form of digital code. Perhaps the most developed of these is SealClub [45] which adds a QR code to each document which specifies an encrypted "reference version". The authenticating user scans a short video of their printed media and then the system computes a homography between the two documents and reports areas of significant visual inconsistency. While not focused on legal process, this approach has many similarities to our own, including the choice to reflect inconsistencies for the user to evaluate. However, the most important difference is that their approach requires an online reference copy of the source document, which is unavailable in our context as criminal process documents are *de facto* sealed at the time of service.

Finally, closest to our own design is a short system proposal by Dlamini *et al.* [18], which similarly suggests using OCR to capture textual features of documents and embed them in cryptographically secure barcodes. However, their design assumes fixed-location content fields, and, as near as we can tell, this proposal was never fully fleshed out, let alone built and tested. Thus, we are unable to compare it to our system.

9 Limitations

WarrInt is a prototype and, as such, is a first cut at providing practical integrity protection for criminal process. Our current approach has a number of limitations and open problems.

OCR Imperfections. Since our system uses OCR, we inherit the limitations of OCR. For example, OCR can confuse certain characters and may not detect punctuation like periods in scanned documents. In principle, an attacker could exploit such ambiguities to bypass the system. However, to do so would require the attacker to identify an existing (and contemporary) piece of legal process where the identifier is almost exactly that of their target, differing only in an easily confused glyph. This is further complicated by the fact that most legal process is not unsealed until considerably after its period of enforceability has expired. This approach may be more feasible, however, for modifying an email address used to receive returns.

Incremental Deployment Vulnerabilities. We have designed our system to be incrementally deployable, because this is a practical necessity of the decentralized nature of US Federal Courts. Thus, districts that make use of *WarrInt*'s enhanced integrity protection would be able to offer greater assurance to providers, but districts using today's methods would be no worse off than they were previously—the two can seamlessly co-exist. However, an inherent side-effect of incremental deployment is that attackers can evade the protections offered by *WarrInt* simply by forging process only from districts that have not yet adopted the system.[21]

Similarly, an attacker can strip away barcodes from a process document or create a forged document without any barcodes. In a more benign setting, the appended bar-code pages could simply be lost. If the courts are incrementally deploying our solution, then it is likely they might service the process document without the accompanying barcodes. In general this is an unavoidable issue but it persists only during the incremental deployment phase of the solution. If individual courts, or even magistrates, were to adopt our solution consistently (i.e., that they use barcodes for all process they issue after some date certain) then it becomes straightforward to detect violations of policy: either the issuing party could report the date they started using *WarrInt* (e.g., via their certificates), or providers could infer it based on the first such process document they received from that issuer.

Increased Workload on Court Staff. While we have tried to minimize the work required by courts and providers, we acknowledge that we still add to their burdens. For example, our system requires that IT staff manage certificates and the integration of *WarrInt* into CM/ECF and that clerks and judges are trained to understand the purpose of the appended barcode page (i.e., that they are not mistakes and should not be erased or overwritten). Similarly, providers must train analysts to use *WarrInt*'s ability to highlight inconsistent text and understand their role in identifying those inconsistencies that have semantic impact on returns. In future work, we hope to conduct a user study to understand the usability of our approach for both providers and courthouse staff.

Managing Certificates. Today, criminal legal process has no inherent integrity protection—the bar is low. Thus, even coarse-grained integrity guarantees (i.e., via a district-wide court certificate) represent a dramatic improvement over the *status quo*. However, should such a system be adopted, it is natural to move towards finer-grained credentials and, with them, the added complexity of their management. It is an open question what is the best tradeoff in credential granularity (i.e., per-judge, per-clerk, per-courtroom, etc.) versus usability and management overhead. Moreover, as the number of private keys increases, the risks of key compromise and the practical challenges of key revocation may increase as well. While outside the scope of this paper, we are well aware that such PKI administration questions have resisted easy answers.

10 Discussion

WarrInt has been designed to provide significant improvement in integrity, subject to the practical constraints of the U.S. Federal legal system. However, significant fractions of legal process served on U.S. providers arises from state courts.[22] It is an open question if our approach would operate equally well in the state context or would require modification. However, there is reason to be optimistic since most states avail themselves of authorities delegated to them under the Federal Stored Communication Act (to bypass jurisdictional issues when serving out-of-state corporations) and, in our perusal

[21] This is particularly true because unlike *physical* seizure warrants, which must issue from the district in which a search will take place, criminal process typically used to acquire data (i.e., subpoenas, 2703(d) orders, and warrants under the Stored Communication Act) all may issue from any district.

[22] Precisely how much of the criminal process handled by providers originates from state authorities is unclear since we are unaware of any comprehensive database of state judicial statistics, similar to those maintained by the Federal court system. However, as one datapoint, Twitter reports that between July of 2020 and July of 2021 (the last time such transparency data was released), just over 29% of their government legal requests were from state authorities, with the remainder being Federal. [65]

of several state search warrants, they remain quite similar in content and structure. However, such an evaluation across the set of 50 states, U.S. territories, and tribal courts, is beyond the scope of our current effort.

More generally, absent the constraints of a particular legal process, one might design a very different system. For example, the European Union's eEvidence Regulation provides a standardized approach to serve criminal process (European Production Orders, or EPOs) directly on Internet service providers across member states[23] that represents a more comprehensive,[24] "top-down" style design in line with traditional applied cryptographic best practice. As part of this process, the e-Evidence Digital Exchange System (eEDES)—now subsumed into the Justice Digital EXchange (JUDEX)—is charged with providing a secure inter-state network for exchange of EPOs, which includes the federated validation of cross-border digital certificates (themselves standardized under the E.U.'s eIDAS signature regulation). Thus, a European magistrate can sign an EPO with a Qualified Electronic Signature (QES), using a certificate provided by an E.U.-authorized Qualified Trust Service Provider (QTSP) and this mechanism can be embedded within Adobe's existing digital signature framework (i.e., a recipient can validate a document's QES using Adobe's built-in tools) [13,22,50]. While the eEvidence Regulation is set to enter into effect August 18, 2026, there is considerable scrambling to support its edicts (only five of the 27 member states were compliant with their statutory obligations by February 2026 [13]).

In principle, the U.S. could pursue similar mechanisms but there are few reasons to believe that such changes are likely to happen soon. The combination of the U.S.' particular common law system, the independence of its judiciary, the highly decentralized nature of U.S. courts, and the limited interest of the U.S. Congress, all create an environment that resists coordinated technological change. For example, from the introduction of Rule 41 in 1946, it took over thirty years before it was modified to allow remote/telephonic swearing, and another 34 years before electronic service was deemed allowable. In 2000, the Federal ESIGN act made electronic signatures carry the same legal weight as ink signatures, but the courts were exempted from this change [1]. The Federal Court system was almost entirely based on paper records until 2002, when the CM/ECF system was rolled out. Written primarily in Perl, the system has been repeatedly added to and patched—complicated further by the fact that each district frequently adds its own unique patches and features to its version of the code base. As well, criminal process has been an area of considerable sensitivity and, until COVID, paper filing of criminal process was the norm in many districts.

[23] Save Denmark, who opts-out of E.U. Justice and Home Affairs policies.

[24] The regulation does not require the use of EPOs; providers receiving traditional Mutual-Legal Assistance Treaty requests or same-country criminal process orders via traditional channels are still obligated to abide by them.

11 Summary

Users have an expectation that the sites they visit will store their data securely and take reasonable efforts to prevent the data from being stolen by criminals. However, even if a service's technical security is perfect, forged legal process represents a backdoor that bypasses virtually all such protections. To the extent they can comply, providers under a court's jurisdiction are generally obligated to produce any and all responsive documents, yet they have no principled way to ascertain if a request they receive is authentic, authorized, and unchanged from when it was issued. Indeed, it is clear that criminal actors are increasingly aware of this issue and multiple attacks via forged legal process have been documented. We have identified the constraints on implementing an easily-deployable solution for the Federal court system and designed a system that abides by those constraints—providing practical integrity protection and reliably detecting forgeries.

Acknowledgments

We thank Miro Haller, Anish Ukani, Shambhavi Mittal, Vrishan Inukollu, Porfirio Montoya, Sathvik Balakrishna, Kanaad Deshpande, Sean Zadig, and our reviewers. We indebted to the current and past law enforcement agents, prosecutors, judges, court clerks, and provider compliance experts who informed the design of *WarrInt*. Special thanks to Seamus Hughes for his invaluable guidance regarding PACER and the chief judges of the 43 district courts that approved our exemption request. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-2038238. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Appendices

A Ethical Considerations

There are a number of stakeholders implicated by our work—both in its goals and methods. The first group includes those involved in requesting and serving legal process (i.e., criminal investigators and prosecuting attorneys) as well as those in the judicial system who oversee the granting of such authorities. These stakeholders' interests are naturally aligned with the goal of improving legal process integrity. Similarly aligned are the legal staff of service providers, who would benefit from an independent mechanism for validating that the legal orders they receive are authentic and properly authorized. Finally, potential victims stand to benefit by having their personal information better protected against such process forgery [10].

The primary ethical concerns around our methods are 1) that our technical discussion might inform attackers how to better forge legal process and 2) that the concrete examples we provide might implicate privacy interests on behalf of those named. With respect to the first concern, we believe that the balance of interests here very much favors publication. There is ample evidence that legal process forgery is already being exploited in the wild and yet we are unaware of significant efforts to improve the status quo, or even identify the practical challenges in doing so. We also take care not to describe the full set of details that would need to be properly emulated to create a "perfect" forgery (since they are not germane for evaluating our approach). As to the second concern, all of the documents we use are public records and readily available to anyone via the Federal PACER system (as described below in our Open Science commitment).

B Open Science

We have provided the code for our system, which should be readily executable by other researchers. We also provide three example documents in various forms throughout the *WarrInt* process: canonical form with a document description embedded in a barcode, scanned unmodified form, digitally modified form, and scanned modified form. For each form, we show how *WarrInt* annotates potential modifications to the document. We choose two representative documents (one from the train dataset and one from the test), and a worst-case document for spurious inconsistencies.

All of the source documents we use in our analysis were obtained from the PACER system, which provides a public portal to the Case Management/Electronic Case Files (CM/ECF) system used by the U.S. Federal courts. While the PACER system is open to the public, it normally charges US$0.10 per page for any documents downloaded. To defray this cost, we applied for, and received, a payment waiver from a broad array of U.S. district courts.[25] However, a condition of that waiver is that we are not allowed to share the documents we obtained in this manner with third parties.

We have addressed this sharing limitation in two ways. First, for a subset of documents, we paid for them independently (i.e., not subject to the waiver) and those we include in their entirety. These can be used to test and validate that the system behaves as expected. For the remaining documents used in our analysis, we have included a detailed list to allow researchers to duplicate our dataset themselves. For each district court, we provide a list of document identifiers of the form O:YY-TT-NNNNN, where O is the divisional office code, YY is the year of filing, TT is the type of case (typically mj for magistrate, but sometimes cr or mc), and NNNNN is the docket ID. The O:YY-TT-NNNNN identifier, entered into PACER for the particular district court, is sufficient to identify the associated docket which, in turn, will contain the individual process documents (e.g., warrant request, signed warrant, return, etc) available to the public.

All of these materials, including source code, documents, and document identifiers can be found at https://doi.org/10.5281/zenodo.20386813.

[25] See the "Fee Exemption for Researchers" page at https://pacer.uscourts.gov/my-account-billing/billing/fee-exemption-request-researchers to request a similar waiver.

References

[1] 15 U.S. Code § 7003 - specific exceptions.

[2] 18 U.S. Code § 2703 - required disclosure of customer communications or records.

[3] Adobe. Adobe PDF Extract API. https://developer.adobe.com/document-services/docs/overview/pdf-extract-api/.

[4] Amazon Web Services. Amazon Textract. https://aws.amazon.com/textract/.

[5] American Civil Liberties Union. Open Whisper Systems Subpoena Documents. https://www.aclu.org/sites/default/files/field_document/open_whisper_documents_0.pdf, October 2016.

[6] Apple. CSAM Detection. https://web.archive.org/web/20210817013304/https://www.apple.com/child-safety/pdf/CSAM_Detection_Technical_Summary.pdf, August 2021.

[7] Jim Axelrod and Andy Bast. CBS News investigation finds fraudulent court orders used to change Google search results. *CBS News*, July 2019.

[8] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded Up Robust Features. In *Proc. of ECCV*, May 2006.

[9] Hanna Bloch-Wehba. Exposing Secret Services: A First Amendment Right of Access to Electronic Surveillence Orders. *Washington Law Review*, 93(1), March 2018.

[10] Jon Brodkin. Verizon fell for fake "search warrant," gave victim's phone data to stalker. *Ars Technica*, December 2023.

[11] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: Binary Robust Independent Elementary Features. In *Proc. of ECCV*, September 2010.

[12] Coherent Graphics Ltd. Coherent PDF. https://www.coherentpdf.com/.

[13] European Commission. Commission takes action to ensure complete and timely transposition of EU directives. https://ec.europa.eu/commission/presscorner/detail/en/inf_26_679, March 2026.

[14] Amanda Cox. QR Code Size: Minimum Size, Maximum Size & More. https://tritonstore.com.au/qr-code-size/, February 2023.

[15] Sagnik Das, Kunwar Yashraj Singh, Jon Wu, Erhan Bas, Vijay Mahadevan, Rahul Bhotika, and Dimitris Samaras. End-to-end Piece-wise Unwarping of Document Images. In *Proc. of IEEE/CVF ICCV*, October 2021.

[16] Antigone Davis and Guy Rosen. Open-Sourcing Photo- and Video-Matching Technology to Make the Internet Safer. https://about.fb.com/news/2019/08/open-source-photo-video-matching/, August 2019.

[17] Aboudramane Diarra, Tegawendé F Bissyande, and Pasteur Poda. Doc-Patch: An Unsupervised Approach for Documents Forgery Detection. In *Proc. of IEEE ACAI*, December 2024.

[18] Nelisiwe Dlamini, Sthembile Mthethwa, and Graham Barbour. Mitigating the Challenge of Hardcopy Document Forgery. In *Proc. of IEEE icABCD*, August 2018.

[19] Brian Donahue. Understanding Law Enforcement Email Compromise (LEEC). Kodex Blog, March 2026.

[20] Matt Donahue. Securing the Front Door to Legal is Now Mission Critical. Kodex Blog, March 2026.

[21] Charles Doyle. Administrative Subpoenas in Criminal Investigations: A Brief Legal Analysis. *Congressional Research Service*, December 2012.

[22] EVIDENCE2e-CODEX. https://evidence2e-codex.eu/.

[23] exiftool. https://manpages.org/exiftool.

[24] Facebook. Law Enforcement Online Requests. https://www.facebook.com/records/login/.

[25] Federal Bureau of Investigation. Easy Access to Information for Conducting Fraudulent Emergency Data Requests Impacts US-Based Companies and Law Enforcement Agencies. https://www.ic3.gov/CSA/2024/241104.pdf, November 2024.

[26] Federal Rules of Criminal Procedure. https://www.uscourts.gov/file/78324/download.

[27] Alan Feuer and Eli Rosenberg. Brooklyn Prosecutor Accused of Using Illegal Wiretap to Spy on Love Interest. *New York Times*, November 2016.

[28] Jonathan Frankle, Sunoo Park, Daniel Shaar, Shafi Goldwasser, and Daniel Weitzner. Practical Accountability of Secret Processes. In *Proc. of USENIX Security*, August 2018.

[29] Marcel J. E. Golay. Notes on digital coding. *Proceedings of the IEEE*, 37:657, 1949.

[30] Google. Google Transparency Report. https://transparencyreport.google.com/user-data/us-national-security.

[31] Google. Law Enforcement Request System. https://lers.google.com/.

[32] Apurba Gorai, Rajarshi Pal, and Phalguni Gupta. Document fraud detection by ink analysis using texture features and histogram matching. In *Proc. of IEEE IJCNN*, July 2016.

[33] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[34] Vrishan Inukollu, Porfirio Montoya, Ross Greer, Alisha Ukani, Katherine Izhikevich, Earlence Fernandes, Stefan Savage, and Alex C. Snoeren. Countering Fabricated Legal Documents: Aligning Verification Attention to Salient Objects through Zero-Shot Detection of Confounding or Obscuring Symbols. In *Proc. of PP-MisDet Workshop at IEEE CVPR*, June 2026.

[35] Adam Jatowt, Mickael Coustaty, Nhu-Van Nguyen, and Antoine Doucet. Deep Statistical Analysis of OCR Errors for Effective Post-OCR Processing. In *Proc. of ACM/IEEE JCDL*, June 2019.

[36] Kodex. https://www.kodexglobal.com/.

[37] Brian Krebs. Hackers Gaining Power of Subpoena Via Fake "Emergency Data Requests". *Krebs on Security*, March 2022.

[38] Brian Krebs. Crooked Cops, Stolen Laptops & the Ghost of UGNazi. *Krebs on Security*, September 2024.

[39] Joshua Kroll, Edward Felten, and Dan Boneh. Secure Protocols for Accountable Warrant Execution. https://www.cs.princeton.edu/~felten/warrant-paper.pdf, 2014.

[40] Enze Liu, Gautam Akiwate, Mattijs Jonker, Ariana Mirian, Stefan Savage, and Geoffrey M. Voelker. Who's Got Your Mail? Characterizing Mail Service Provider Usage. In *Proc. of ACM IMC*, November 2021.

[41] David G Lowe. Object Recognition from Local Scale-Invariant Features. In *Proc. of IEEE ICCV*, September 1999.

[42] Brendan J. Lyons. Search Warrants are rarely unsealed. Here's why. *Times Union*, August 2022.

[43] Ke Ma, Zhixin Shu, Xue Bai, Jue Wang, and Dimitris Samaras. DocUNet: Document Image Unwarping via A Stacked U-Net. In *Proc. of IEEE CVPR*, June 2018.

[44] Amir Markovitz, Inbal Lavi, Or Perel, Shai Mazor, and Roee Litman. Can You Read Me Now? Content Aware Rectification using Angle Supervision. In *Proc. of ECCV*, August 2020.

[45] Ochoa Martín, Vanegas Hernán, Toro-Pozo Jorge, and Basin David. SealClub: Computer-aided Paper Document Authentication. In *Proc. of IEEE ACSAC*, December 2023.

[46] Meta. Transparency reports. https://transparency.meta.com/reports/.

[47] Microsoft. Microsoft Law Enforcement Request Portal. https://leportal.microsoft.com/home.

[48] National Consortium for Justice Information and Statistics. SEARCH. https://www.search.org/.

[49] Katelyn Polantz and Aileen Graef. Federal courts go old school to paper filings after hack to key system. *CNN*, August 2025.

[50] Jorge A. Espina Ramos. European Preservation and Production Orders: A Non-Exclusive Approach to E-Evidence within the EU. *eucrim*, (3), 2025.

[51] Rod J. Rosenstein. Policy Regarding Applications for Protective Orders Pursuant to 18 U.S.C. § 2705(b). https://www.justice.gov/criminal/criminal-ccips/page/file/1005791/dl?inline, October 2017.

[52] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Proc. of ECCV*, September 2006.

[53] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *Proc. of IEEE ICCV*, November 2011.

[54] Tim Sparks. Wyden bill would require digital signatures for sensitive court orders. *Cyberscoop*, July 2021.

[55] Tesseract. https://github.com/tesseract-ocr/tesseract.

[56] Richard M. Thompson II. The Fourth Amendment Third-Party Doctrine. *Congressional Research Service*, June 2014.

[57] Twibright. OptAR. https://ronja.twibright.com/optar/, 2016.

[58] U.S. Courts. Administrative Office of the US Courts: Forms. https://www.uscourts.gov/forms-rules/forms.

[59] U.S. Courts. Statistical Tables for the Federal Judiciary - M3 Tables. https://www.uscourts.gov/statistics-reports/caseload-statistics-data-tables.

[60] U.S. Courts. Judiciary Addresses Cybersecurity Breach: Extra Safeguards to Protect Sensitive Court Records, January 2021.

[61] U.S. District Court, District of Connecticut. Search Warrant Case Opening Instructions. https://ctd.uscourts.gov/sites/default/files/forms/MJSearchWarrantInstructionsforUSAOasof04.26.13.pdf, April 2013.

[62] U.S. General Services Administration. Federal PKI Governance and Compliance Audit Information. https://www.idmanagement.gov/fpki/.

[63] Verizon. Verizon Transparency Reports. https://www.verizon.com/about/investors/transparency-report.

[64] James Williams. The Unofficial Guide to Facebook's Law Enforcement Portal Version 2. https://cdn.netzpolitik.org/wp-upload/2016/08/facebook-law-enforcement-portal-inofficial-manual.pdf, April 2014.

[65] X. X Transparency Center. https://transparency.x.com/en.

[66] Yahoo. Government Data Requests. https://www.yahooinc.com/transparency/.

C Additional Background on Legal Process

Here we briefly review the most common forms of Federal criminal legal process served on third-parties, as well as the legal and procedural requirements for obtaining them and ensuring secrecy.

C.1 Types of Process

We proceed roughly in order of commonality (i.e., how frequently such orders are served), starting with process for retrospective data (i.e., existing data that is stored) and then prospective data requests (i.e., for live data in transit).[26]

[26] We are specifically ignoring emergency data requests (EDRs) in this paper, a mechanism by which a law enforcement agency may directly request information under the "exigent circumstances" doctrine, based on a representation that imminent death or serious bodily injury is at stake. While EDR forgery is a real problem [37], because it operates outside the formal legal process context, it may require a distinct solution.

Subpoenas. In the Federal system, an impanneled grand jury empowers a prosecutor (typically an Assistant US Attorney) to issue a subpoena demanding retrospective records from third parties as part of an investigation. The standard for requesting a subpoena is quite low, requiring no more than "official curiosity" (United States v. Morton Salt Co., 338 U.S. 632, 652 (1950)). The power to compel via subpoena is far-reaching and, absent specific protections embodied in statute or prior case law, extends to virtually any existing records that a party may possess that fall within the scope of the request.[27] Unlike other forms of process, a grand jury subpoena is not reviewed or signed by any judge or magistrate.[28]

2703(d) Orders. 18 U.S.C. 2703 documents a portion of the Stored Communications Act that controls government access to retrospective "stored communications" (e.g., e-mail, voice mail, text messages). This statute specifically protects such information from blanket access via subpoena[29] but, via 2703(d), provides that a specific court order (colloquially called a "d-order") can be used to obtain any "record or other information pertaining to a subscriber to or customer of such service (not including the contents of communications)"—i.e., metadata. This includes information such as the "To:" and "From:" fields in e-mail (but not the "Subject"), sender and recipient accounts for DMs, etc. The standard for obtaining such an order is that the government demonstrate "specific and articulable facts showing that there are reasonable grounds" to believe that the records sought are "relevant and material to an ongoing criminal investigation" [2].

SCA Warrants. The same section of the Stored Communications Act, (particularly, 18 U.S.C. 2703(a) and (b)) also authorizes the government to obtain the retrospective contents of stored communication from a provider (e.g., the text of a subscriber's email) via a warrant. Federal warrants are governed by Rule 41 of the Federal Rules of Criminal Procedure and require a demonstration (as per the 4th amendment) of "probable cause" to believe that specific evidence would be obtained of violations of a particular Federal crime. As described earlier, this demonstration is typically via a sworn affidavit provided by a Federal law enforcement agent.

Note, while most warrants must be issued in the district in which a search or seizure is to take place, 2703 creates an exception ("any court of competent jurisdiction") which allows SCA warrants in particular to issue from *any* district.[30]

PRTT Orders. The Pen/Trap Statute (18 U.S.C. 3121-3127) allows Federal courts to authorize electronic surveillance concerning *prospective* "dialing, routing, addressing, or signaling information transmitted by an instrument or facility from which a wire or electronic communication is transmitted". While the precise mapping of this telephone-era statute to Internet services can vary, it is generally metadata (i.e., To: and From: fields in e-mail, sender and recipient accounts for DMs, etc). The standard for obtaining such an order is that the government represent that "the information likely to be obtained is relevant to an ongoing criminal investigation." Roughly speaking, a PRTT order can be thought of as the prospective version of a 2703(d) order.

Wiretaps. The Wiretap Act, as codified in 18 U.S.C 2510-22, regulates the government's *prospective* surveillance on the content of "wire, oral, or electronic communications". This can include live interception of audio on voice calls, text content for SMS, live e-mail content, or even live packet capture for a particular address. As per 18 USC 2518, Wiretap requests (colloquially called T3's because of their origin in Title III of the Omnibus Crime Control and Safe Streets Act of 1968) require the same probable cause requirements as a warrant plus a range of additional limitations and constraints, as well as oversight from the court (including, via 18 U.S.C. 2519, the public reporting of all wiretap order requests and metadata about their execution). Wiretap orders themselves are sealed by statute and rarely are made public.

Overall, subpoenas, 2703(d)-orders and warrants (i.e., the requests for retrospective data) dominate the legal process served on Internet services. For example, in Google's transparency report for the first half of 2024, 89% (54,474 of 61,402) of the requests for user data received from US government entities fell into one of these categories [30]. There are similarly high percentages in the most recent transparency reports from Yahoo (90%), Verizon (70%) and Meta (67%)[31] [46, 63, 66].

C.2 Obtaining Process Documents

As described earlier, a Federal grand-jury subpoena is created by the requesting prosecutor and specifies the recipient (for a company, this may be the official "custodian of records") and what records or testimony are sought.[32] The final docu-

[27] There are a range of grounds on which a recipient may seek to challenge a subpoena, but it is worth highlighting that the combination of widespread outsourcing to cloud services (e.g., for e-mail service [40]) and the long-standing "third-party doctrine" (i.e., that by providing data to a third party one typically surrenders existing "reasonable expectation of privacy") [56] means that individuals may be unaware of subpoenas concerning their information and would lack standing to challenge them even if they were aware.

[28] For certain classes of crimes and for certain information, law enforcement can directly issue *administrative subpoenas*, independent of any court, with similar powers to compel [21]. Notably, 18 U.S.C. 2703 specifically empowers administrative subpoenas authorized in Federal or state statute.

[29] Indeed, 2703 limits the power of subpoenas to "basic subscriber information" such as the user's identity, contact info and billing information.

[30] Were this not the case, the districts in the Bay Area (cand), Seattle (wawd), and Virginia (vaed) would be overwhelmed handling out-of-district warrants for the rest of the country.

[31] Meta's smaller percentage is due to the large numbers of PRTT orders they received, which we surmise may be due to their operation of WhatsApp.

[32] One common practice is to subpoena a corporation for testimony concerning records and include a cover letter indicating that documents can be provided "in lieu" of testimony [5].

ment includes the signature of the issuing court's clerk and the seal of the court (although we understand that these are commonly pre-filled) and specifies the requesting prosecutor, who typically initials it as well. As we understand it, current practice in most districts does not involve recording copies of criminal grand jury subpoenas, and they generally do not appear in CM/ECF unless they are litigated or are attached to a related request (i.e., most commonly for a 2705 preclusion of notice order). Thus, in the context of our system, ensuring integrity for grand jury subpoenas would likely require each U.S. Attorney's office to adopt a version of *WarrInt*.

All other forms of process require court oversight to obtain and generally require sworn affidavits from law enforcement officers attesting to the facts supporting the relevant evidentiary standard.[33] It is common practice that the management of most Federal criminal process is delegated to Magistrate Judges (MJs), who are appointed to eight-year, renewable terms by the district's District Judges. These judges rotate "criminal duty" (i.e., so an MJ is available for criminal process requests every day).[34] These magistrates review requests and the associated documentation and then decide whether to issue an order, request changes, or deny the request. The final order will be dated, signed by the judge and (for all but the subpoena) will provide the case number and the dates of validity. A paper copy of the process may be provided in court to the sworn agent, but it is common that the clerk makes an electronic copy (possibly scanned) available to the requesting prosecutor, who, in turn, e-mails it to the investigating agent.

Some forms of issued third-party process involve standard forms—typically based on those issued by the Administrative Office of the U.S. Courts. Notably, most grand jury subpoenas appear to use some version of form AO110, and search warrants some version of AO93 (or, for tracking warrants, AO103). Others, such as 2703(d) orders and PRTT orders, have no standard form and are rendered by the judge in prose. In all cases, the particular identification of the information to be provided is typically contained in attachments after the order (e.g., Attachment B for search warrants) although it may also be listed in the order itself.

C.3 Process Secrecy

Most regular court orders are public by default, however investigatory criminal process is a key exception. Documents produced to the grand jury are presumptively *under seal* (a by-product of grand jury secrecy rules) and thus such subpoenas and their returns are rarely ever made public (typically, they appear in public records only in response to specific unsealing litigation or because the receiving party publishes it directly in the media). Similarly, wiretap orders are sealed by default via statute. For the rest—including 2703(d) orders and SCA warrants—they are also commonly sealed, either based on local district rules that stipulate their default sealing or via explicit sealing requests included with a request for a court order.[35] Once a sealing order is granted, it may be for indefinite duration, and such documents are then only unsealed via a specific court order.[36] Overall, sealing is very much the norm for criminal process and our searches of PACER only identify a small fraction of the corpus of warrant requests that are quantified in corporate transparency requests.

Non-disclosure is somewhat more complex since it implicates a third party's First amendment right to speech and thus requires a specific court order. In particular 18 U.S.C. 2705(b) provides for a "preclusion of notice" order which may be requested from the court along with a 2703(d)-order, an SCA warrant or a subpoena.[37] The recipient of such an order is prohibited from disclosing the existence of the government's request to "any other person". To obtain an order under 2705(b), the government must represent that a significant "adverse result" would occur if the request for information were disclosed (including endangering life or safety, flight from prosecution, tampering with evidence, intimidation of witnesses, etc.) It is unclear what the practical time limits are for such 2705 orders and historically many appear to have had unlimited duration. However, as a result of a lawsuit settled with Microsoft, since 2017 the official Department of Justice policy is that absent "exceptional circumstances" such preclusion should only last for a year [51].

D Document Alignment Using OCR features

Our system seeks to align two copies of the same document, each of which may be a photographed, scanned, or digital document. Moreover, one copy (the service provider's) may or may not have alterations. On the assumption that both documents are planar, traditional methods of aligning such documents involve estimating the homography matrix $\mathbf{H}$, which represents a projective transformation that relates the coordinates of points between two images of the same planar surface [33]. It is a 3×3 matrix defined as:

$$\mathbf{p}' = \mathbf{H}\mathbf{p}$$

[33] Historically, government affidavits have been sworn in person, before the presiding judge handling the request, but post-COVID they are increasingly sworn telephonically.

[34] Wiretap is an exception, as the statute explicitly requires a District Judge's oversight, as are Foreign Intelligence Surveillance Act (FISA) warrants, which can only issue from the FISA Court.

[35] Beyond sealing, a separate level of care applies to so-called "highly sensitive documents" (HSDs), which are not even filed in CM/ECF because of concerns about the system's past compromise and are typically filed exclusively in paper form at the courthouse. [60]

[36] We are aware that some districts have local rules that criminal process is filed under seal by default, but will be unsealed once returns are delivered to the court absent a specific sealing order. We believe that such rules are responsible for a large portion of the Federal search warrants in our dataset.

[37] The provisions of 2705 are only available for process described under Section 2703, but grand jury subpoenas are specifically authorized via 2703(c)(2). PRTT and wiretap orders have their own integral preclusion statutes as well.

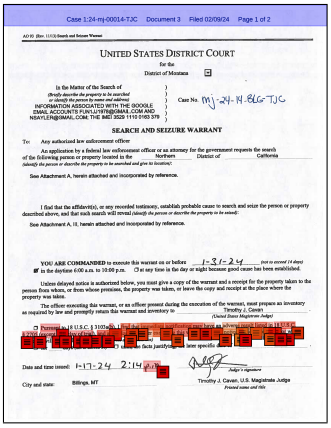


Figure 5: Search warrant with 42 spurious inconsistencies.

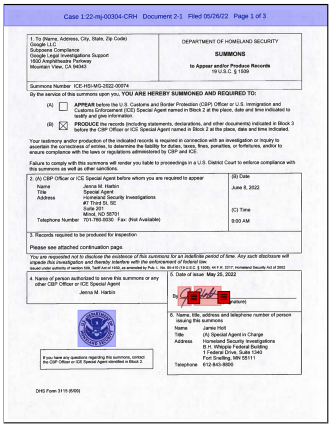
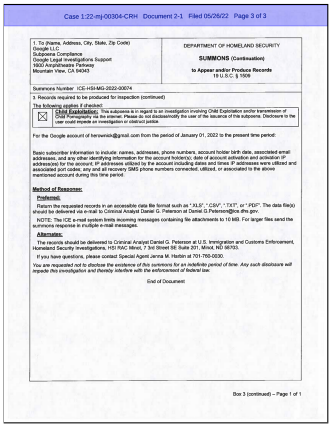


Figure 6: Scanned, unmodified administrative subpoena with two spurious inconsistencies (red).



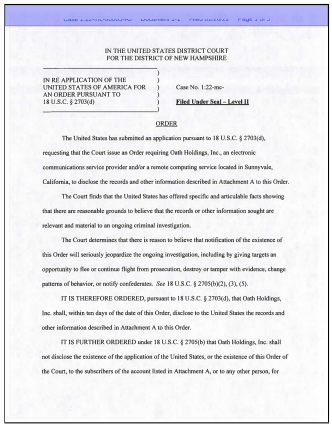
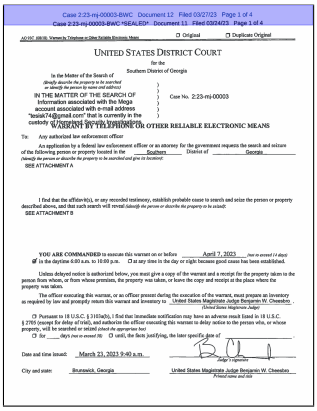
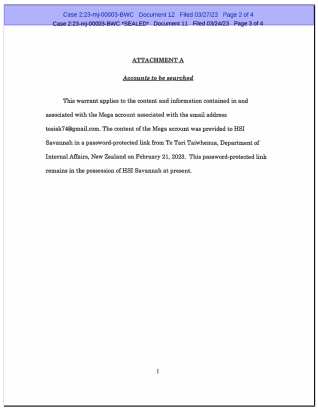
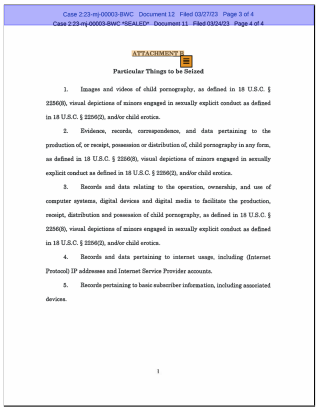


Figure 7: First page of scanned, unmodified 2703(d) order.







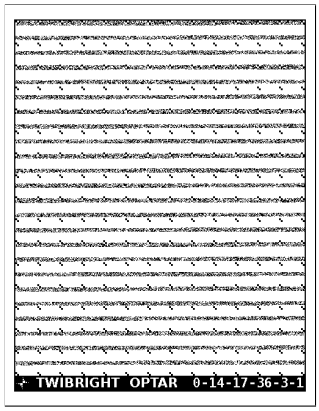


Figure 8: *WarrInt*'s 2D-barcode and output for a scanned, unmodified search warrant with one spurious inconsistency (yellow).

where $\mathbf{p} = [x, y, 1]^T$ and $\mathbf{p}' = [x', y', 1]^T$ are the homogeneous coordinates of a point in the original and transformed image, respectively. The homography matrix $\mathbf{H}$ is parameterized by 8 independent parameters, as it has 9 elements in total, but the scale factor is typically normalized.

Estimation methods often rely on image-based features to detect and match keypoints between images [8, 11, 41, 52, 53], applied often in both document processing [45] and general computer vision. However, they require access to the original image data which is prohibitively large to transmit to the service provider in our use case. Hence, we use OCR outputs as features for this geometric transformation estimation.

The core of our methodology is replacement of traditional correspondence keypoints and feature descriptors in the homography estimation algorithm, and computation of feature similarity as a string-matching task. Specifically, (1) gradient-based scale-space extrema, or similar keypoint candidates, are replaced by the centroids of OCR-detected words, and (2) high-dimensional gradient-information descriptors are replaced by the OCR-extracted textstring. Using matched word pairs, we compute a homography matrix that maps one document's coordinate system to the other. Since OCR outputs may contain noise and outliers, we employ random-sampling consensus (RANSAC) to robustly estimate the homography. RANSAC iteratively selects random subsets of matches, computes the homography, and evaluates its consistency with the remaining matches. This process ensures that the final homography is robust to OCR measurement and string recognition errors and outliers. We note that further alignment techniques exist for cases where folding, crumpling, or other non-linear transforms affect the document correspondences [15, 43, 44], which may be adapted to this use case.

E Example *WarrInt* Output

In Figure 5, we show an example outlier where *WarrInt* has a large number (42) of spurious inconsistencies. We also provide another outlier example in our open science repository. In Figures 6 to 8 we show examples of *WarrInt*'s output for different types of canonical documents scanned at DPI 200: an administrative subpoena, a 2703(d) order, and a search warrant that includes the 2D-barcode page that *WarrInt* generates.