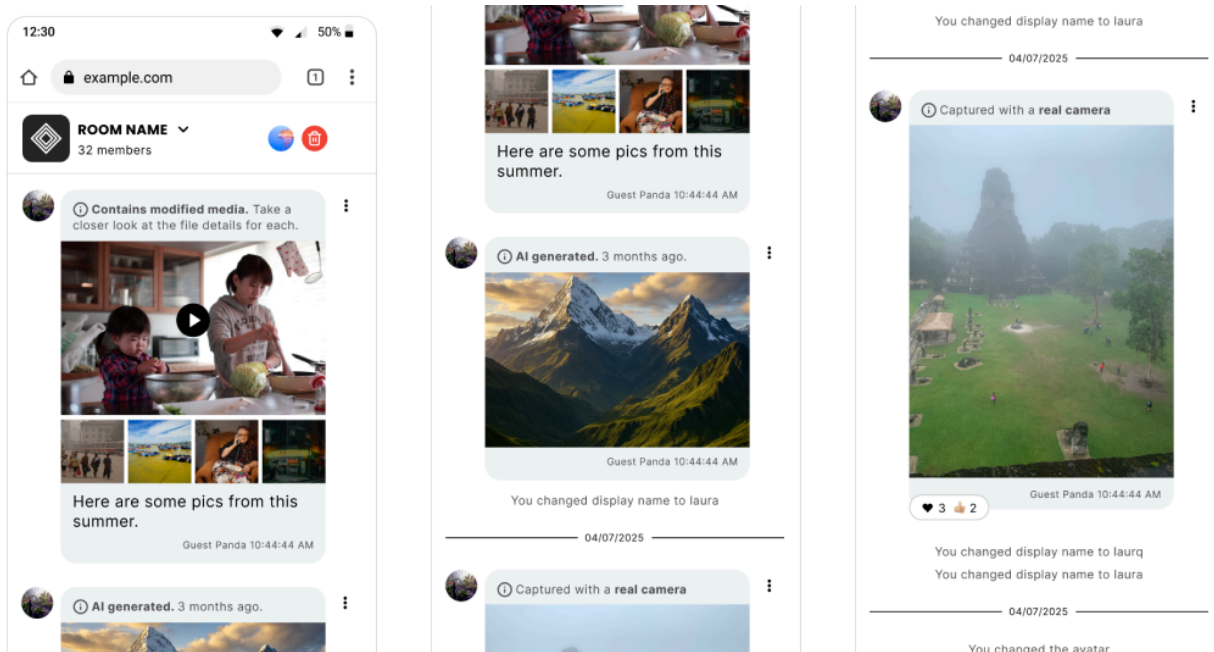


Enhancing Visual Trust in Messaging Apps



Integrating Media Provenance and Authentication User Experiences Into Secure Online Conversation Spaces

October 2026

Jointly developed by [Guardian Project \(Proofmode\)](#), [Okthanks](#) and [Rights Action Lab \(RAL\)](#)

1. Executive Summary

Messaging apps have become the front line of visual evidence. When a crisis unfolds, the images and videos that reach journalists, human rights defenders, and diaspora communities rarely arrive through newsrooms or verified archives. They arrive in a chat thread, forwarded by a relative or a trusted contact, stripped of context and often of every piece of data that could establish where they came from.

At the same time, the cost of producing convincing falsehoods has collapsed. Generative AI can fabricate a plausible disaster scene in seconds, and authentic footage from years past can be recirculated as breaking news with nothing more than a new caption. Recipients are left to judge authenticity by eye, which is exactly the judgment these techniques are designed to defeat.

This paper describes how Guardian Project and RAL integrated [Proofmode](#) and [C2PA Content Credentials](#) into a browser-based secure messaging app, to close that gap. RAL has been instrumental in Proofmode's development and in evaluating its real-world implementation by providing partial funding, leading user testing to ensure Proofmode is user-friendly, and testing this integration within the communication tool. Our thesis is simple: **a secure messenger should use every available signal, from cryptographic Content Credentials down to basic file heuristics, to tell recipients what is known about a piece of media, how confidently it is known, and what cannot be verified at all.** Rather than issuing opaque trust verdicts, the application should expose evidence and equip people to reach their own conclusions.

Key contributions

- **A layered signal model** that combines C2PA manifests, Proofmode proof data, conventional file metadata, and heuristics, each labeled with an honest assurance level.
- **Five in-line interventions** (Captured with a camera, Modified, Generated with AI, Older than 3 months, Screenshot) that translate technical provenance into plain-language cues inside the message thread.
- **Room-aware privacy defaults** that protect at-risk senders in group settings while preserving full-fidelity evidence in direct conversations.
- **Provenance surfaced on both sides of the send button**, so senders see and control what they share, and recipients see what can and cannot be verified.

2. The Problem: Trust Breaks Down in the Thread

Public discussion of synthetic media tends to focus on sophisticated deepfakes. In practice, the misinformation that moves through messaging groups during a crisis is usually far simpler, and it succeeds because the channels it travels through erase the clues that would expose it. We see three recurring patterns.

Resurfaced historical media

Authentic images and videos from past events are routinely re-shared during current crises to support false narratives. Footage of South African firefighters from prior years, for example, repeatedly resurfaces on social platforms and in messaging groups framed as a present-day emergency response. Nothing about the pixels is fake; only the implied date and place are. Detection tools that look for manipulation will not flag it.

AI-generated synthetic content

Generative tools now produce realistic but entirely fabricated scenes, such as the viral animal-rescue clips that circulate during wildfire seasons. This material has no genuine capture history. It may carry subtle synthesis artifacts, but those are rarely visible on a phone screen and grow harder to spot with each model generation.

Stripped metadata and screen captures

Mainstream messaging apps re-encode media and discard metadata in transit, largely for bandwidth and privacy reasons. Recipients who forward content often do so via screenshots, which replace the original file with a new one that carries no history at all. By the time evidence reaches the person who needs to evaluate it, every signal of origin has usually been removed (Figure 1).

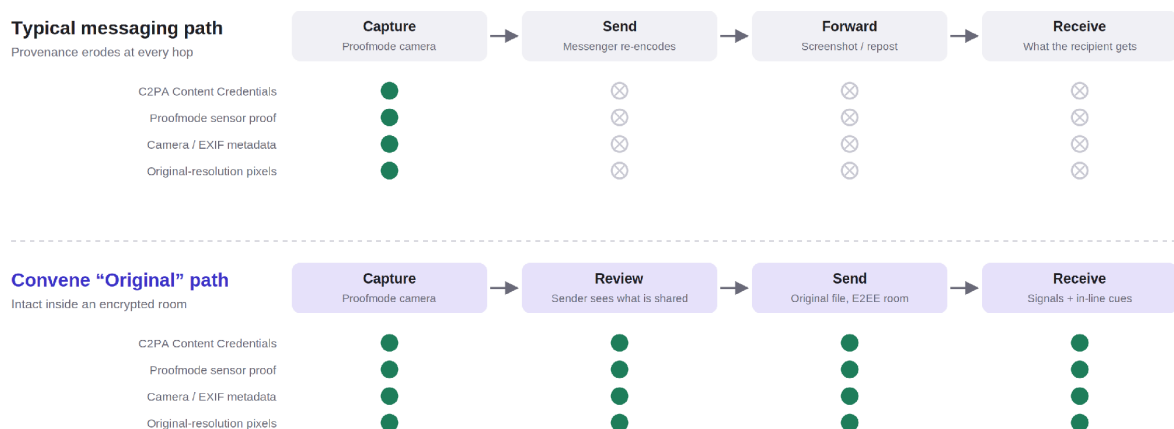


Figure 1. In a typical messaging path, provenance signals are lost at the first hop. When a sender chooses Original, the file and all of its provenance arrive intact inside an encrypted room.

3. Design Principles

Designing provenance features for people in hostile environments means holding three goals in tension: evidence must be trustworthy, the people who create it must stay safe, and the interface must be usable by non-experts under stress. Four principles guided our work across the teams at Guardian Project, Okthanks, and RAL.

Harm reduction for at-risk groups. Raw metadata can endanger lives. Precise GPS coordinates, device serial numbers, or even a phone model can help an adversary identify a source.

Transparency mechanisms must establish the integrity of content without exposing the identity of its creator.

Context-sensitive exposure. What is appropriate to reveal depends on where media is shared: the size and composition of a room, the threat models of its participants, and the risk that content is seen by unintended third parties.

Privacy-preserving defaults. Vulnerable users should be protected without any configuration. Privacy holds unless someone deliberately chooses otherwise, and that choice is always visible and reversible.

Progressive assurance. Cues must clearly distinguish cryptographically verified provenance from unverified metadata and from simple heuristics, without discrediting the weaker signals, which are often all a recipient has.

4. Architecture: A Layered Signal Model

RAL's messaging app is a secure chat platform co-designed with human rights advocates and digital security trainers, with rooms that are private and encrypted by default. The provenance features described here build on more than fifteen years of Guardian Project work on secure communication and verifiable media, including the Proofmode camera, key strategic and funding support and extensive user testing provided by RAL, and on the open C2PA standard for Content Credentials.

We approached the integration through specific lenses: members of a diaspora receiving evidence from their community, journalists seeking eyewitness accounts from sources, and human rights advocates gathering verifiable records for their research. Through testing conducted with RAL, the goal was not only to support a standard, but to give those recipients practical tools to tell what is real. That led us to treat provenance as a stack of signals rather than a single yes-or-no check (Figure 2).

The signal stack: every layer contributes, each with an honest assurance level



Figure 2. The app reads every available layer of evidence and labels each with the level of assurance it can actually support.

Cryptographic layers. A C2PA manifest is a signed record embedded in the file that can describe how it was created (for example, captured by a camera or generated by a model) and what edits were made. Proofmode adds signed hashes, device and sensor metadata, and a capture timestamp. Either can be checked for tampering.

Descriptive layers. Conventional EXIF, XMP, and IPTC metadata often records the camera model, capture date, and editing software. It is valuable context, but it can be edited by anyone and so cannot be treated as proof without a digital signature.

Heuristic layer. Even when metadata is gone, the file itself still says something: dimensions that match a phone screen, an aspect ratio, a format, or evidence of prior compression.

Preserving the evidence in transit. When a sender chooses Original quality, the app transmits the unmodified file, so its embedded Content Credentials and metadata travel with it. The Compressed option re-encodes the image and removes that data, which is the right default in some contexts (Section 6). In end-to-end encrypted rooms the server cannot see the media, so extracting and interpreting these signals happens on the client only.

5. In-Line Interventions

Content Credentials only help if people see them. In a messaging app, that means provenance must be visible at two moments: before media is sent, so the sender understands and controls what will be shared, and after it arrives, so the recipient can weigh it in context.

Beyond designing a schema for displaying Content Credentials in chat, we combined them with the other signal layers to produce short, plain-language cues we call interventions. Each intervention answers one question a recipient is likely to have. We selected five, based on the kinds of evidence

diaspora communities are actually receiving today (Figure 3).

From raw signals to one plain-language cue

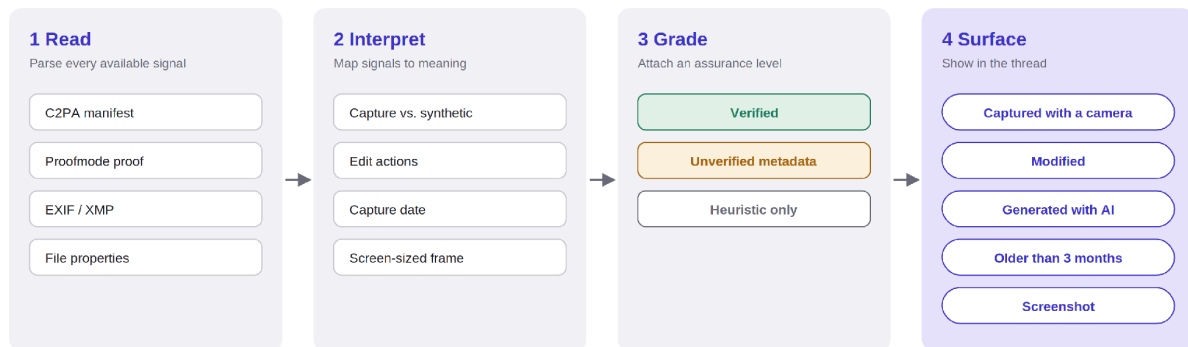


Figure 3. Signals are read, interpreted, graded for assurance, and surfaced in the thread as a single, readable label.

Intervention	What it tells the recipient	Strongest signal	Fallback signal
Captured with a camera	The media originated from a physical camera, not a generator or a screen.	C2PA capture assertion or a Proofmode proof with sensor data	Camera make, model, and exposure data in EXIF
Modified	The media has been changed since it was captured.	Edit actions recorded in the C2PA manifest; hash mismatch against a Proofmode proof	Editing-software tags in file metadata
Generated with AI	The media was produced wholly or partly by a generative model.	C2PA digital source type declaring AI generation	Generator-written metadata, where present
Older than 3 months	The media was captured well before it is being shared.	Signed capture timestamp (C2PA or Proofmode)	Original capture date in EXIF
Screenshot	The media is a capture of a screen, not of the world.	A declared screen-capture source type	Screen-matching dimensions and absent camera data

Because interventions draw on the whole signal stack, they work with or without Content Credentials. A screenshot with no metadata can still be labeled as a screenshot; a file with a valid manifest earns a stronger, verifiable label. The five were chosen to target the patterns in Section 2 directly: “Older than 3 months” addresses resurfaced media, “Generated with AI” addresses synthetic content, and “Screenshot” and “Modified” flag the lossy, recontextualized copies that dominate forwarded threads.

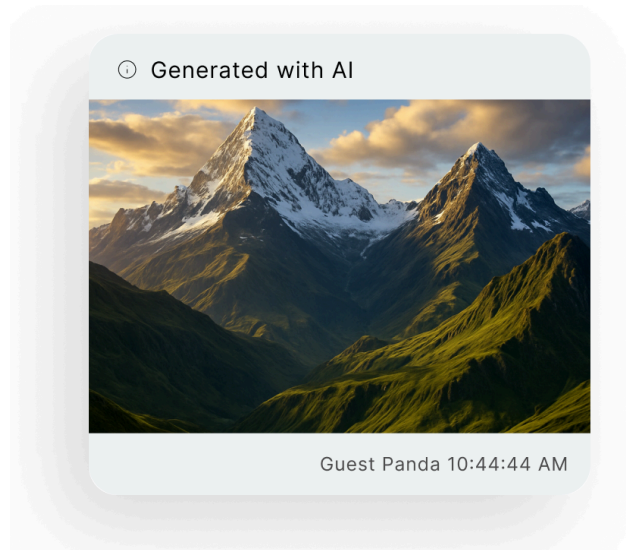


Figure 4. Interventions give recipients insight in the message thread, with or without Content Credentials.

6. Context-Aware Privacy Defaults

Metadata can expose people. We had to deliver media transparency without revealing anything about a sender that they had not knowingly chosen to share. Our design research made the stakes concrete: communicators in the diaspora were uneasy even about showing the model of the phone that took a picture.

In Witness’ published research on [“C2PA Content Credentials and the Surveillance Risk”](#), they point out a risk, where a potential user documents something that could put them in danger. They ask that the image not be published under their name, but the embedded media credential still carries his name and other device identifiers. Photos and their credentials shared in messaging apps will eventually travel through different systems. What is meant to be used and verified privately may need to be stripped of information in certain cases when used publicly.

We resolved this by making defaults depend on the room (Figure 4):

1. **Group chat rooms** continue to compress images and withhold media details unless the sender explicitly chooses to share them.
2. **Direct message rooms**, where evidence is typically exchanged and trust is higher, default to sending the original file and its metadata whenever Content Credentials are present.

In both cases, interventions remain visible as a hint that more information exists, which raises awareness of provenance features without forcing disclosure.

Room-aware defaults: exposure follows the trust model of the conversation

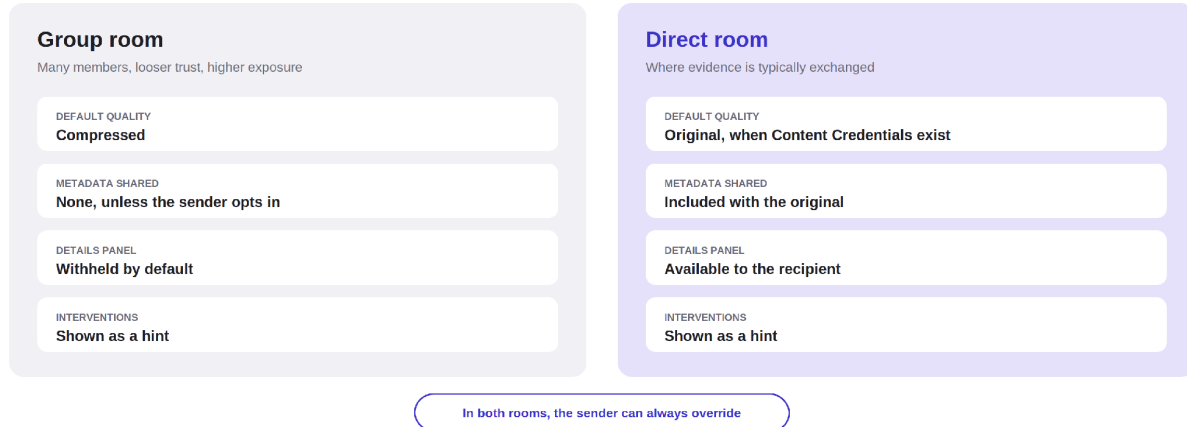


Figure 5. Defaults follow the trust model of the conversation, and the sender always has the final say.

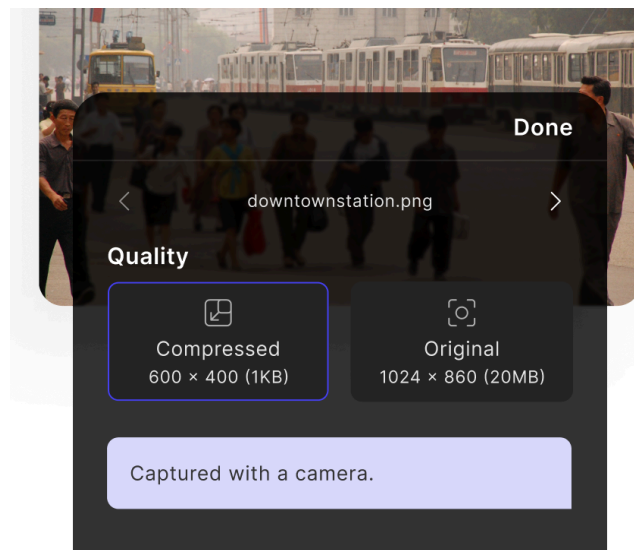


Figure 6. The quality sheet makes the trade-off explicit: Compressed protects the sender, Original preserves the evidence.

7. Progressive Assurance

Because the messaging app user experience surfaces everything that can be known about a piece of media, it must also be candid about how much of it can be trusted. Ordinary metadata may have been altered; Content Credentials let a recipient confirm that it has not. The interface needs to make that distinction clear without implying that unverified information is false.

We explored several models of assurance levels and ultimately chose restraint: short, neutral messages that state whether the details shown can be verified, and point to where verification can happen (Figure 5). We deliberately avoided alarm-style warnings and numeric trust scores, which invite recipients to outsource their judgment rather than exercise it.

Progressive assurance: say what is known, and how well it is known

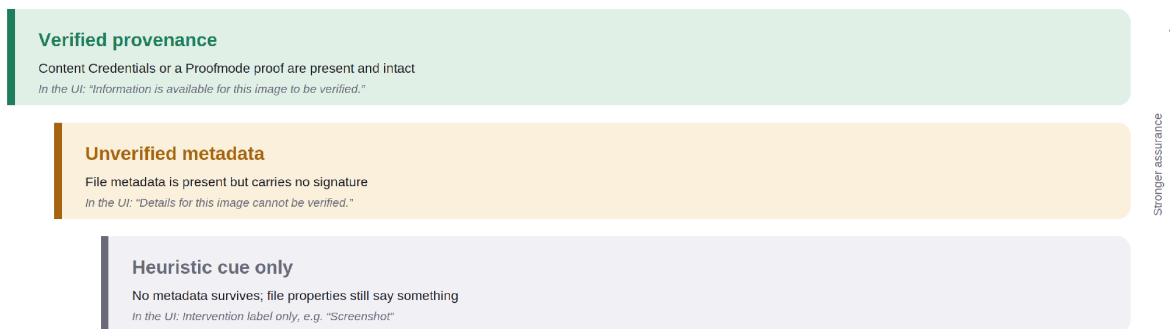


Figure 7. Three tiers of assurance, each paired with the message the recipient actually sees.

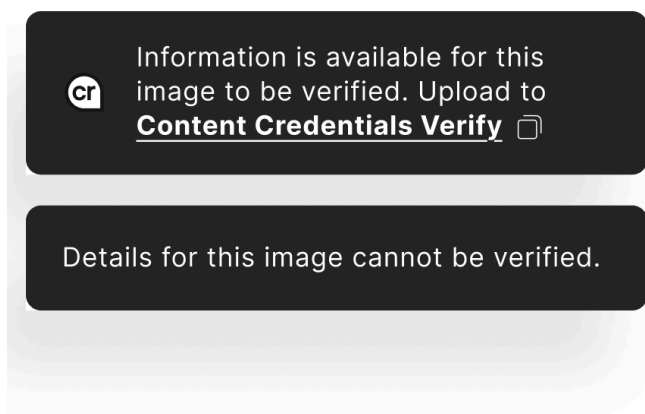


Figure 8. Subtle, neutral messages tell recipients whether details can be verified.

8. The End-to-End Experience

Taken together, these elements form a single flow that begins before the send button and continues after delivery (Figure 6). The sender sees exactly what will be shared and chooses the right trade-off for the room. The recipient sees a cue in the thread, can open the details, and is told plainly whether those details can be independently verified.

Provenance is surfaced twice: before sending, and after receiving

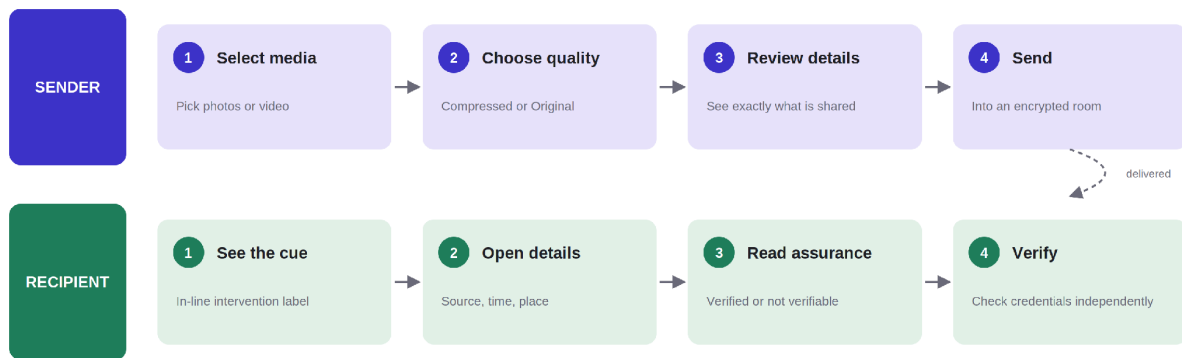


Figure 9. Provenance appears at each step of the sender and recipient journeys.

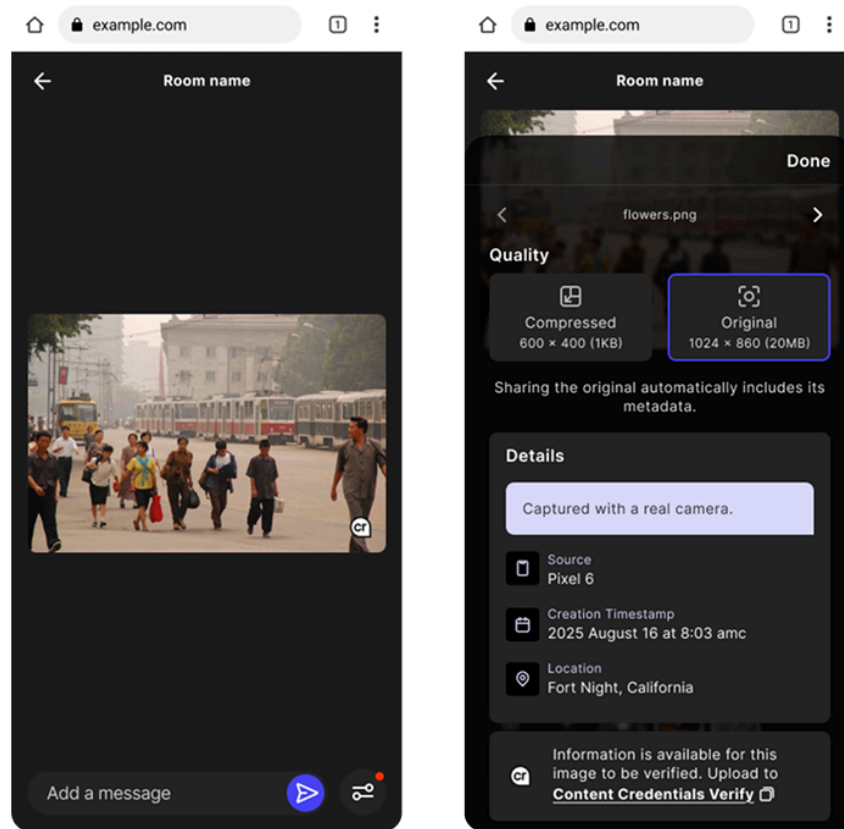


Figure 10. The sender flow: selecting media, attaching it to a message, and reviewing quality and provenance details before sending.

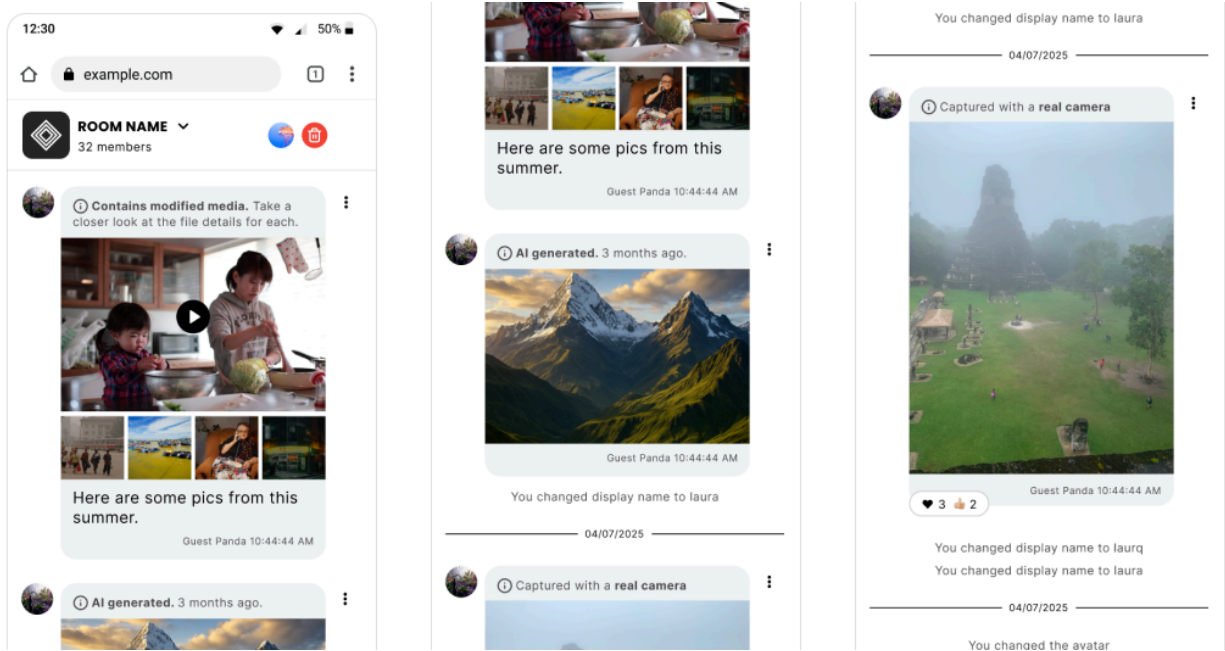


Figure 11. *Interventions in context: modified media, AI-generated media, and camera captures as they appear in a room.*

9. Limitations and Open Questions

We want to be explicit about what this approach does not do.

- **Absence of provenance is not evidence of fakery.** Most authentic media today carries no Content Credentials. Interfaces must never treat a missing manifest as a negative signal, and ours is designed not to.
- **Heuristics can be gamed.** File dimensions and metadata can be forged. That is why heuristic and descriptive signals are always labeled as unverified.
- **Credentials prove origin, not truth.** A verified camera capture can still be framed misleadingly. Provenance narrows the question; it does not answer it.
- **Verification should not leak the evidence.** Today, deeper verification relies on uploading media to a web-based Content Credentials verifier. For sensitive material, verification that happens entirely on the device is preferable.
- **Adoption is uneven.** The value of cryptographic layers grows as more cameras, editors, and generators sign their output.
- **Support for storage and bandwidth constrained users.** Sharing of “original” media without compression can be difficult for users connecting through devices with limited storage and low bandwidth connections.

10. Conclusion and Outlook

By embedding Proofmode and C2PA Content Credentials directly into the secure messaging app workflow, we have shown that media provenance can be both accessible and privacy-preserving. Clear, contextual cues before and after media is sent help people navigate hostile information environments with confidence, while room-aware defaults ensure that transparency never comes at the expense of the people creating evidence.

The patterns described here are not specific to this one app. We believe they offer a practical model for any messaging platform that wants to support visual trust:

1. Preserve provenance end to end whenever a sender chooses to, ensuring it is also very clear how to opt-out.
2. Combine every available signal, and label each with the assurance it can actually support.
3. Translate provenance into a small set of plain-language cues shown in the thread.
4. Let the context of the conversation, not a global setting, decide what is exposed by default.
5. Inform judgment rather than replacing it.

Guardian Project, Okthanks and Rights Action Lab welcome collaboration with platform developers, standards bodies, and civil society partners who share these goals.

About Guardian Project

Founded in 2009, Guardian Project is a global collective of designers, trainers, researchers, and developers building open-source software that enhances privacy and security for everyone. Its work includes Orbot, Proofmode, and ButterBox. Learn more at guardianproject.info.

About Okthanks

Okthanks are product designers, researchers and advocates shaping technology for a freer, safer world. They partner with technologists, human rights organizations and advocates to advance human and digital rights. Learn more at <https://okthanks.com>

About Rights Action Lab

Rights Action Lab (RAL) has been instrumental in the development of Proofmode and its integration into secure communication tools. By providing critical funding, conducting extensive user testing to ensure Proofmode remains intuitive and user-friendly, and evaluating in-app integrations, RAL helps empower users and communities to establish visual trust in secure online spaces. Learn more at rightsactionlab.org.