PNNL-SA-185580

Real-Time Video Authentication using the Double Ratchet Algorithm

Jake Benz[1], Marie Whyatt[1], Bill Nickless[1], Cullen Tollbom[1], Richard Griswold[1]
Camille Palmer[2], Michael Rosulek[2], Gayathri Garimella[2], Jaspal Singh[2]

[1]Pacific Northwest National Laboratory (PNNL), Richland, WA, USA
[2]Oregon State University (OSU), Corvallis, OR, USA

Email: Jacob.Benz@pnnl.gov, Marie.Whyatt@pnnl.gov, Bill.Nickless@pnnl.gov, Cullen.Tollbom@pnnl.gov, Richard.Griswold@pnnl.gov, palmecam@oregonstate.edu, rosulekm@engr.orst.edu, garimelg@oregonstate.edu, singjasp@oregonstate.edu

Acknowledgement: This work is supported by the U.S. Department of Energy, National Nuclear Security Administration, Office of Defense Nuclear Nonproliferation Research and Development.

Abstract

This project addresses a shortcoming present in many other authentication solutions in international arms control and safeguards: key management and security, which is the critical lynchpin in any data authentication scheme and a significant challenge. The protocol presented in this paper is unique in that encryption keys are updated with every message generated through an initial key agreement protocol, the Diffie-Helman key exchange protocol, and a hash function to create new keys. Current authentication approaches rely on secret or public key encryption, both of which have strengths and weaknesses. The encryption key update with every message exchange means that a compromised key cannot be used to decrypt previous or future messages.

This capability is why the protocol is termed “self-healing”. Future remote verification activities will generate lots of data, which is central to generating evidence of treaty compliance, and therefore it is important that the data be trusted. “Self-healing” encryption, used within the Signal and WhatsApp secure messaging apps, can greatly increase the confidence in and security of future verification equipment, for both attended and remote regimes. This paper will present ongoing research towards the implementation of a surveillance system using this approach, to demonstrate a proof-of-concept solution for near or real-time data authentication capability.

Keywords: Video Authentication, Arms Control, Cryptography, Double-Ratchet, HMAC [1]

PNNL-SA-185580

Overview and Mission Need

Future arms control treaty regimes are likely to cover all aspects of the nuclear weapon lifecycle. This includes locations and processes that an inspecting party may not have access to. This may be due both to the sensitive nature of the location or process, e.g., dismantlement, as well as the ongoing and dynamic nature of those processes. A typical solution to address this challenge is to employ chain of custody measures, consisting of a combination or subset of seals, unique identifiers, surveillance, and facility inspection. The goal is to establish and maintain a boundary of control around the item, process, or infrastructure in the absence of inspectors.

A robust chain of custody tool is surveillance. The continuous and visual nature of surveillance makes it function much like an additional inspector watching over an item, area, or process. The challenge with surveillance is the same as with inspector presence, how will the host party assure itself that sensitive data is not being released to the inspecting party? One option is for the host party to have the right to review surveillance images prior to releasing them to the inspecting party and be able to redact images as necessary. This approach puts the burden on the inspecting party to maintain confidence in the images that have been in host control and potentially some redacted prior to inspector review.

This paper describes a solution to allow for host review and redaction of surveillance images while maintaining inspecting party confidence over image integrity. The solution digitally signs surveillance images in real-time with a unique key for each image to minimize the risk of encryption key compromise.

Introduction to the Double Ratchet Algorithm

Authentication is the process by which the inspecting party develops and maintains confidence in verification equipment and data. Key management and data security are the critical lynchpins in any data authentication scheme, and an ongoing, significant challenge. If not handled correctly it can create a false sense of security and completely corrupt confidence in the data. In the context of surveillance for arms control, if a host party has acquired the encryption keys, then they may be able to insert false or altered images without the knowledge of the inspecting party and which may then be deemed authentic by the inspecting party. To address this issue, the project leveraged an existing protocol, Double Ratchet Algorithm (DRA) [2] which asserts the host party cannot manipulate or substitute previous or future video images, even if they have compromised an encryption key.

DRA is used in secure encrypted communications, including Signal, WhatsApp, and Facebook Messenger to provide confidentiality and integrity of communications between two parties. It is traditionally implemented as a cellphone application and commonly used between two individuals who trust each other and require private communications. DRA offers “backward secret, forward secure” encryption. Each individual message sent is encrypted with its own encryption/decryption keys, generated afresh for each message, hence the term “ratchet”. At a high level, DRA as implemented in Signal, has the following steps for secure messaging between two parties, call them Alice and Bob:

1. Alice and Bob securely initialize a shared secret key through a process called Extended Triple Diffie-Hellman (X3DH) [3].
2. Bob creates a private/public key pair, calculates shared secret key using the Diffie Hellman key exchange protocol, and sends message to Alice encrypted with shared secret key and Bob’s public key.
3. Alice uses Bob’s public key and her private key to calculate shared secret key and decrypts message.
4. When Alice is ready to send a message, she creates a new private/public key pair, calculates a new shared secret key using the Diffie Hellman key exchange protocol, and sends message to Bob encrypted with new shared secret key and her new public key.
5. Bob uses Alice’s new public key and his private key to calculate shared secret key and decrypts message.
6. Steps 2 – 5 are repeated for each message exchanged.

Visually, steps 1-6 are shown in Figure 1 below:

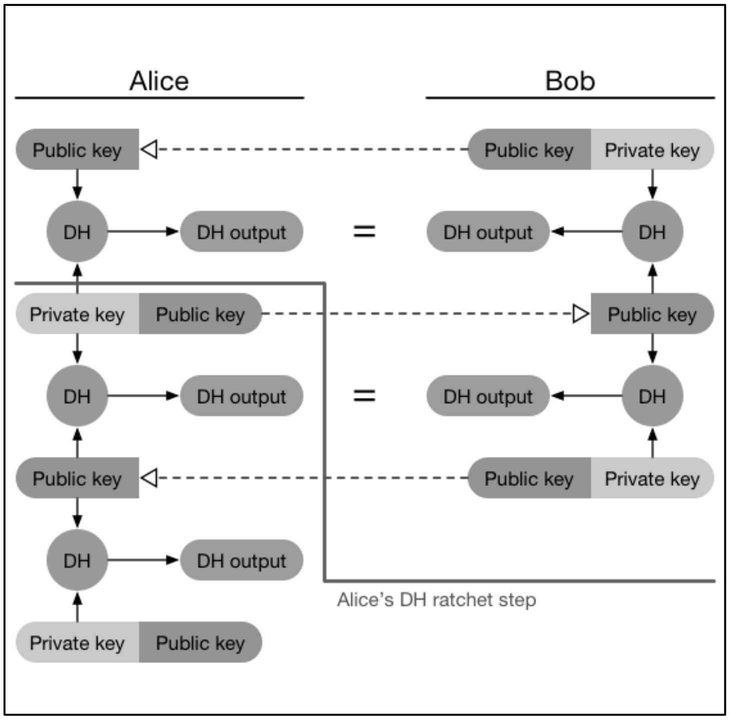

Figure 1: Process for Receipt of Message and Generation of New Keys for Subsequent Message

The feature of DRA which makes it attractive for use in video authentication for arms control is there is no key management needed. Instead, Double-Ratchet algorithm is based on KDF [4] (Key Derivative Function) chains, such that at each “turn of the ratchet” a new public/private key pair is generated and only the public key shared.

PNNL-SA-185580

Modification of the Double Ratchet Algorithm

The Double-Ratchet algorithm was analyzed in 2022 by Cullen Tollbom, who then adapted it to authenticate rather than to encrypt. The modifications focused both on preserving the integrity of the Double-Ratchet algorithm's cryptographic functioning and adapting it with awareness of an arms control working environment and authentication needs. Cryptographic integrity of the modified algorithm was verified independently via a formal proof by experts at Oregon State University which is discussed in the next section.

To authenticate a frame of video in PNG format, an authentication "watermark" consisting of the number of seconds and nanoseconds since the Unix epoch (12:00:00 a.m. UTC, January 1, 1970) and the frame number is written visibly onto a corner of the image for a human to see (Figure 2) and appended digitally to the end of the file. The *chain key* ("DH output" ovals in Figure 1) is then used to compute an HMAC (hash) of the watermarked video frame, and that HMAC is appended digitally to the end of the file after the watermark. The other significant modification is the video frame is not transmitted, but only the watermark. The video frame is instead stored for later review and potential redaction. The inspecting party can use the received watermark and their *chain key* to later authenticate video frames provided after the host has redacted any sensitive information.

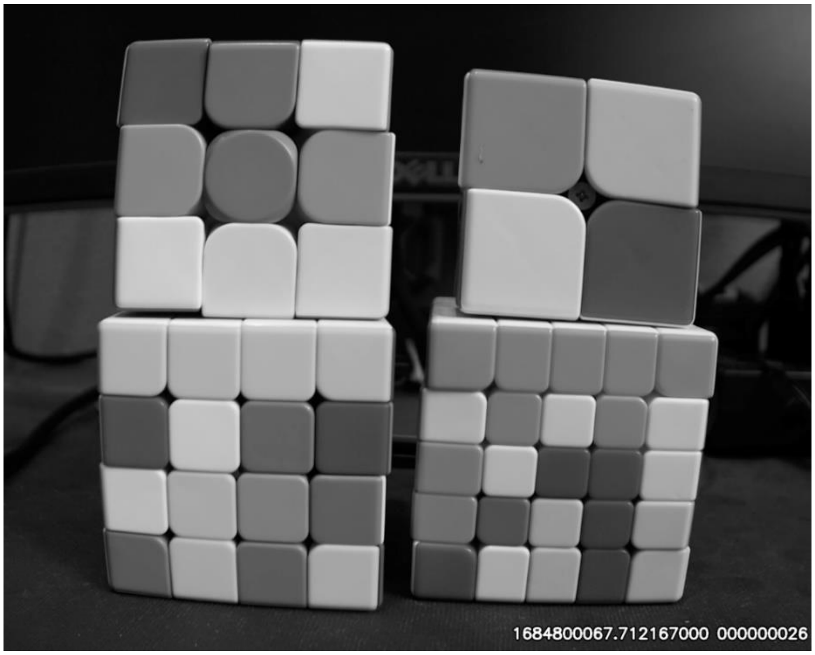

Figure 2: Example watermarked frame with visible watermark at lower right

These modifications were designed to work in an arms control scenario activity in which video from at least one, and likely multiple, cameras are employed. In the scenarios envisioned, the objective is to ensure the integrity of the images while allowing for host party review and redaction prior to release to the inspecting party. Adapting the DRA to focus on authentication, leveraging the security features of continual key updates, and taking advantage of the open-source nature of DRA for transparency support achieving this objective.

A representation of the system architecture with four computing devices are shown in Figure 3. The two shown at left, *Host DRA Device* and *Inspector DRA Device*, are used during the joint activity to collect and authenticate the video frames; both are enclosed in separate tamper indicating enclosures under exclusive control of the Host and Inspector, respectively. The two shown at right, *Host Private Review/Redact workstation* and *Inspector Private Authentication workstation*, are used immediately after the joint activity is done and the video camera(s) stopped.

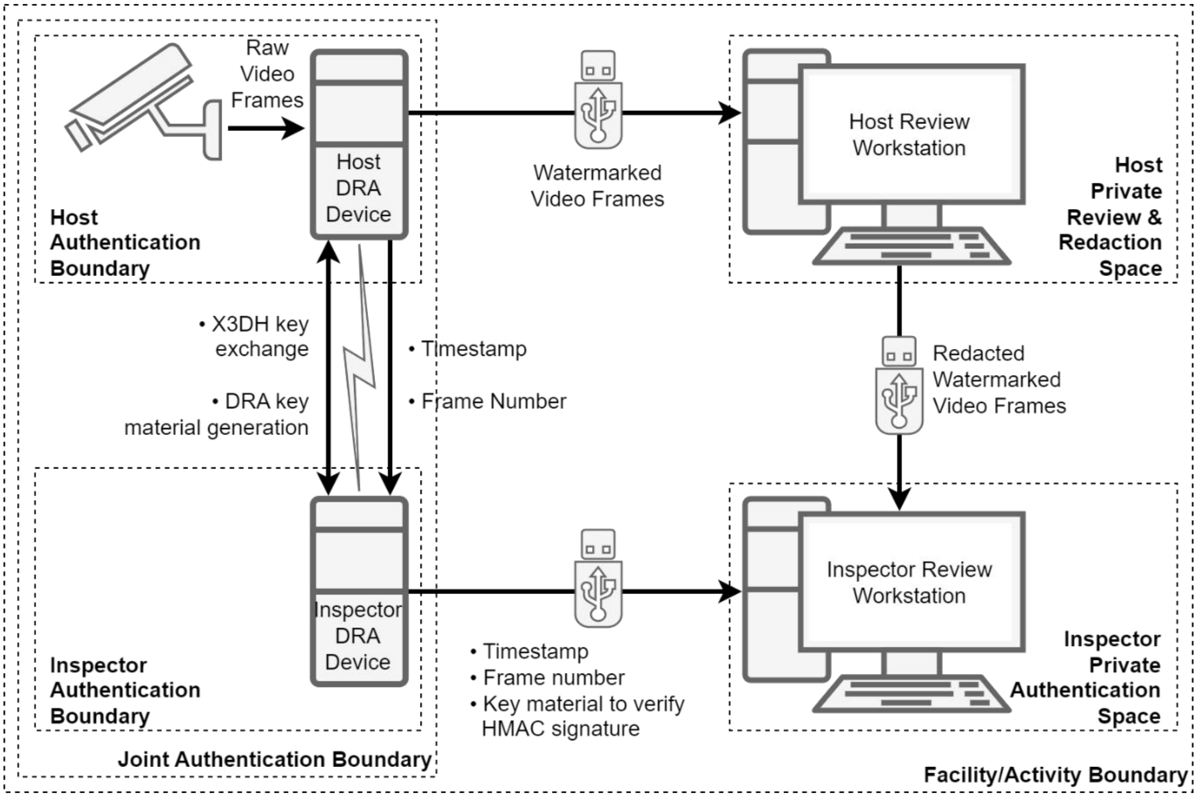
Raw Video Frames
Host DRA Device
Host Authentication Boundary
Watermarked Video Frames
Host Review Workstation
Host Private Review & Redaction Space
• X3DH key exchange
• DRA key material generation
• Timestamp
• Frame Number
Redacted Watermarked Video Frames
Inspector DRA Device
Inspector Authentication Boundary
• Timestamp
• Frame number
• Key material to verify HMAC signature
Inspector Review Workstation
Inspector Private Authentication Space
Joint Authentication Boundary
Facility/Activity Boundary

Figure 3: System architecture of DRA use in an arms control joint activity (single-camera shown)

Prior to the joint activity the *Host DRA Device* and *Inspector DRA Device*, both generate a cryptographic public/private key pair and then exchange their public keys. The exchange can be done in several ways including using electronic media (e.g., ethernet cable, thumb drive, CD ROM) or manually (printed on paper at each device and typed in at the other DRA device using a computer keyboard).

Moments before the joint activity begins, the video camera(s) is started, and the DRA algorithm is launched on both the *Host DRA Device* and the *Inspector DRA Device*. The *Host DRA Device* retrieves a frame of video, turns the “ratchet” producing new key material (public/private keys), transmits the timestamp (e.g., Unix epoch), frame number, and new key material to the *Inspector DRA Device*. The *Host DRA Device* then “stamps” the watermark into the video frame visibly and appends electronically, computes the chain key, uses the chain key to compute the HMAC of the video frame, appends the HMAC, and stores the watermarked video frame on removable media to be retrieved when the joint activity concludes. The *Inspector DRA Device* receives the timestamp and frame number, computes the chain key independently, and stores the timestamp, frame number, and chain key in a file on its removable media to be retrieved when the joint activity concludes. It then turns the “ratchet” and sends its new key material to the *Host DRA Device* for use in computing the next chain key for the next video frame. An HMAC leveraged as the video frame authentication mechanism, is a widely used cryptographic authentication technique using a hash function and a secret key to confirm that data is authentic.

When the joint activity concludes, the Host party opens the tamper indicating enclosure (*Host Authentication Boundary* in) containing the *Host DRA Device*, removes the media containing the unredacted watermarked video frames, takes the media to the *Host Private Review/Redact workstation*, and there uses it in private to view and redact any sensitive watermarked video frames. The remaining redacted video frames are then placed on new electronic media and given to the Inspector party for authentication. Meanwhile the Inspector party opens the tamper indicating enclosure (*Inspector Authentication Boundary* in Figure 3) containing the *Inspector DRA Device*, removes the media containing the file of authentication information, and takes it to the *Inspector Private Authentication workstation* where the information from both removable media are used in private by the Inspector to reconcile and authenticate the redacted video frames.

A physical representation of the system architecture is shown in Figure 4 below. This is the prototype system developed under this project to demonstrate the feasibility of real-time video authentication using the DRA.

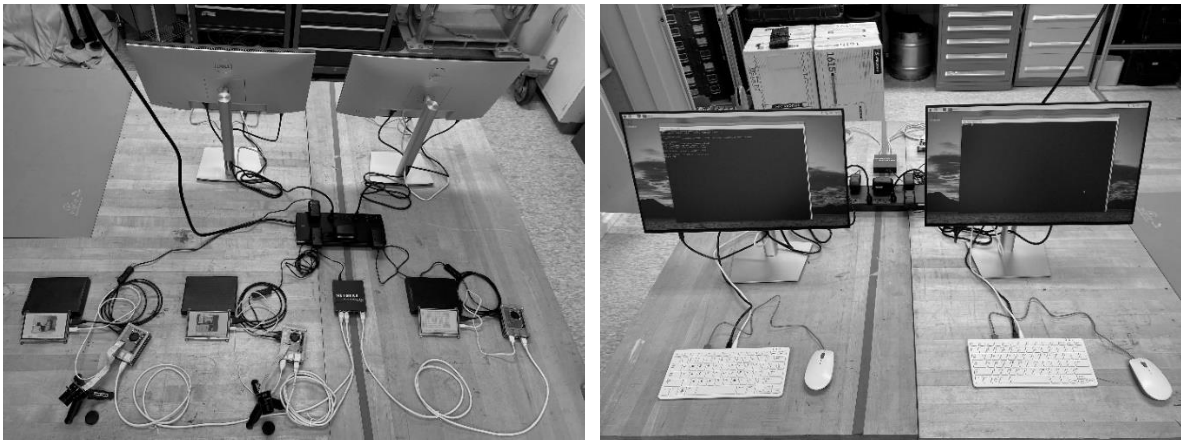

Figure 4: Physical Setup of Two Camera Prototype Video Authentication System

Figure 4 shows the implemented protype depicted by the line drawing in Figure 3. The image on the left shows 2 Host DRA Devices each with a Camera attached (left of tape) and an Inspection DRA Device (right of tape). The image on the right shows the Host Review Workstation and the Inspector Review on the left and right of tape respectively.

Security Proof

After the Double-Ratchet algorithm had been adapted and implementation of the single-camera prototype had begun, the cryptography team at Oregon State University analyzed the adapted algorithm and the operational environment producing a security proof as a measure of rigor meant to increase confidence in the idea and trust in its implementation.

A security proof for encryption is best done as a type of "game". In such a game a finite set of *attack scenarios* is determined, and another finite set of conditions is determined in which one or the other party successfully subverts the encryption and "wins" the game, successfully subverting authentication. either the Host or Inspector

General Steps of a Security Proof Game

At its simplest, provable security has 3 steps:

Step 1) Formulate "attack scenario":

- Victim[s] use certain cryptographic algorithms
- Some inputs are under attacker's control, some are not
- Attacker can see some outputs, others are hidden

Step 2) Cast the scenario as a "security game":

- Attacker interacts with victims
- Attacker can trigger actions by the victims, with attacker-chosen inputs, attacker-visible outputs
- Define a "winning" condition, e.g.:
  - Victim accepts a forgery
  - Attacker can distinguish actual data from random digital bytes

Step 3) System is "secure" if:

- no efficient (poly-time) attacker can win the game with non-negligible (better than 2-n) probability

```
• Init():
    (sk; st_i; st_h) ← setup()
    epoch ← 0
    msg[h->i] = msg[i->h] = empty
• host-adv-ratchet(framevec):
    (st_h; msg[h->i]) ← host-adv-ratchet(st_h; msg[i->h]; framevec)
    epoch++
    frames[epoch] = framevec
    return msg[h->i]
• inspector-adv-ratchet():
    (st_i; msg[i->h]) ← inspector-adv-ratchet(st_i; msg[i->h])
    return msg[i->h]
• leak():
    corr-epoch ← corr-epoch ∪ {epoch, epoch+1, ..., epoch + Δ}
    return st_i; st_h
• finalize(f; mac; idx = (idx1; idx2)):
    win if insp-verify(sk; st_i; f; mac; idx) = true and
    frames[idx1][idx2] ≠ f and
    idx1 ∉ corr-epoch
```

Formalize

Applying (formalizing) these steps to the adapted and implemented Double-Ratchet authentication system requires these assumptions:

- “Victims” are either the Host or the Inspector
- The Host and the Inspector perform the following actions:
    - *init* – initiate the protocol
    - *host-advance-ratchet* – advances the ratchet on *Host DRA Device* and authenticates a frame
    - *inspector-advance-ratchet* – advances the ratchet on *Inspector DRA Device*
    - *host-release* – outputs authenticated frames to the host
    - *host-verify* – verify unredacted frames provided by the host
    - *insp-verify* – verify unredacted frames provided by the host

Define the Security Game

Security Goal: prevent falsification of video frames by “adversarial” Host

Powers of the attacker (Host):

- Observe/Delay all messages along communication channel, and contents of Watermarked Video Frames removable media
- Compromise internal state of Host DRA Device or the Inspector DRA Device (e.g., destroy); but cannot tamper with the contents of the state (e.g., spoof)
- Choose some or all the frames that are input to the camera device

“Winning” condition:

- Adversary causes inspector to accept invalid forgery
    - Forgery = video frame different than the “ground truth” established in step 3 above
    - Invalid = video frame corresponds to a round in which adversary does not know the internal state of the DRA devices

Security Game

- This part of the game captures the adversarial powers
    - Adversary controls the input frames to the camera – input to *host-adv-ratchet*
    - Adversary learns the internal state – *leak* function
- Adversary **wins** if invokes the winning condition in *finalize* function

Security Proof as Game Summary

The contribution of the Oregon State University cryptography team formally specified the modified double ratchet algorithm, proved its security using the game-based framework, reduced security of the scheme to the security of these other well-known cryptographic building blocks/schemes [2]:

- CKA
- PRF-PRNG
- MAC

Interpretation: If an adversary can break our scheme, then it can be used to construct an efficient adversary for CKA or PRF-PRNG or MAC scheme (which are accepted as secure [5]).

The security proof completed by Oregon State University confirmed the original security features of the DRA, backward secure and perfect forward secrecy, remain intact with the modified DRA used in this project.

Conclusion and Next Steps

The project is in the second and last year of the project. Work to date has focused on the modification of the DRA, and its implementation into a prototype system. The prototype system initially consisted of a single camera system to demonstrate early feasibility of the approach. This was successfully demonstrated in January 2023, then focus shifted to extension into a multicamera system. The purpose is a recognition that a deployed system is going to contain multiple cameras, and therefore explore challenges associated with data collection, transmission, and review from multiple independent camera streams. Remaining research will first explore technical and confidence issues with transmitting the data outside of the room where it was generated. This includes a different room in the same facility, or different building onsite. Second, research will cursorily examine design and operational issues to better understand potential implementation and deployment challenges.

References

[1] HMAC - Hash-based Message Authentication Code

[2] T. Perrin and M. Marlinspike. *The Double Ratchet Algorithm.* November 2016. Rev. 1. https://signal.org/docs/specifications/doubleratchet/

[3] M. Marlinspike and T. Perrin. *The X3DH Key Agreement Protocol*. November 2016. Rev. 1. https://www.signal.org/docs/specifications/x3dh/

[4] KDF: Key Derivative Function, a cryptographic algorithm that derives one or more secret keys from a secret value such as a master key, a password, or a passphrase using a pseudorandom function (e.g., hash function, block cipher)

[5] Joël Alwen, Sandro Coretti and Yevgeniy Dodis, "The Double Ratchet: Security Notions, Proofs, and Modularization for the Signal Protocol"