Downgrade prevention for IKEv2

draft-ietf-ipsecme-ikev2-downgrade-prevention

IPSECME — IETF 124

Valery Smyslov, **Christopher Patton**

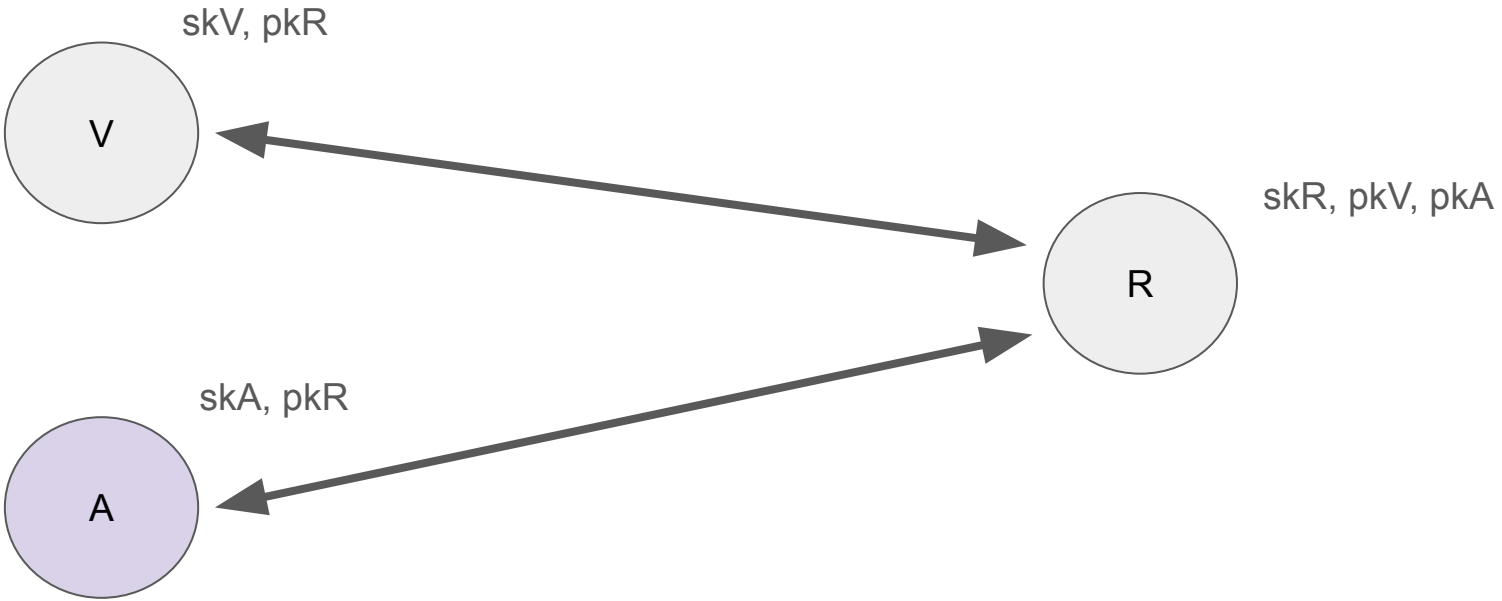
Background

- IKEv2 is vulnerable to ***downgrade attacks:*** If the initiator and responder prefer a **strong** key exchange (KE) algorithm, but both support some **weak** KE, then an active attacker can downgrade the connection to use the **weak** KE.*
    - draft-ietf-ipsecme-ikev2-mlkem et al.: After Q-day, any party that supports both **DH+ML-KEM** and **DH-only** KE is vulnerable.
- Vulnerability: During the initial exchange, each party authenticates its outbound message, but not the inbound message from its peer ⇒ no confirmation that our peer got the right message.
    - draft-ietf-ipsecme-ikev2-downgrade-prevention: Each party authenticates both messages, thereby ***confirming they've had the same conversation***.

* Attacks first described in ia.cr/2016/072, Section 5.

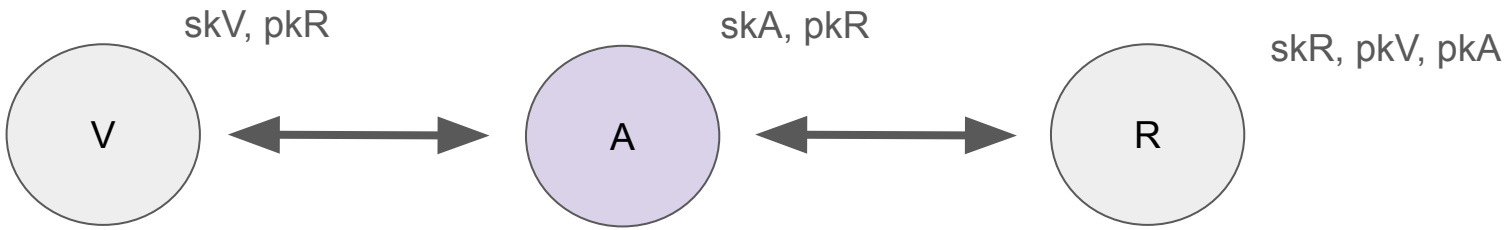
Example attack: Identity misbinding

Prerequisite: Two initiators V and A are configured to speak to responder R using signatures for mutual authentication.



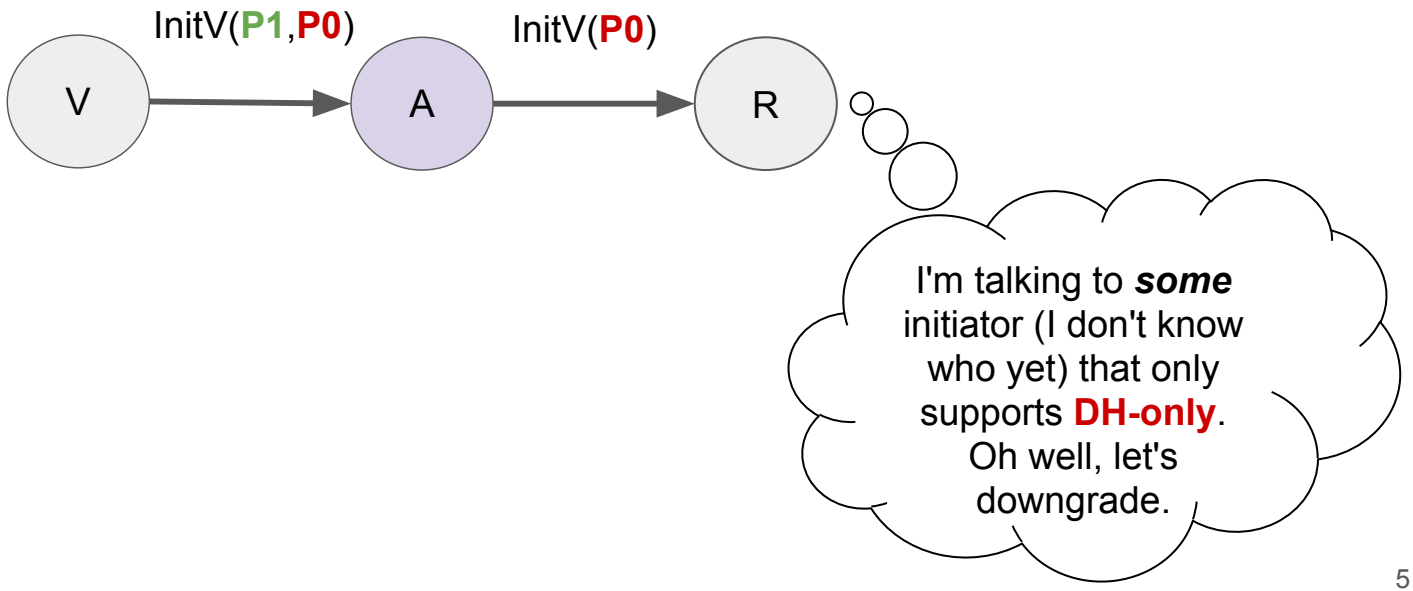
Example attack: Identity misbinding

Setup: A is on-path between V and R and wants to decrypt their traffic.



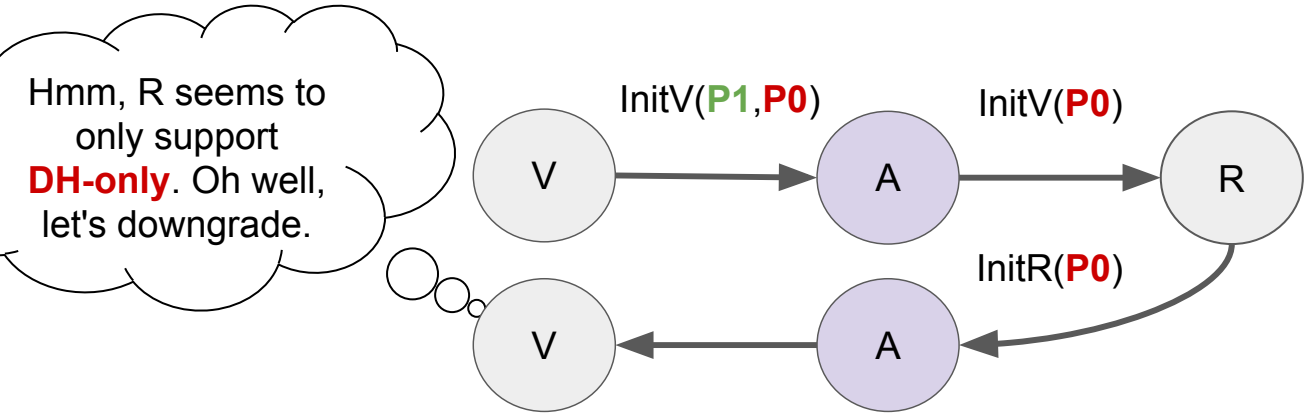
Example attack: Identity misbinding

Step 1: A intercepts V's IKE_INIT_SA message, rewrites it by dropping support for the **strong** KE, and forwards it to R.



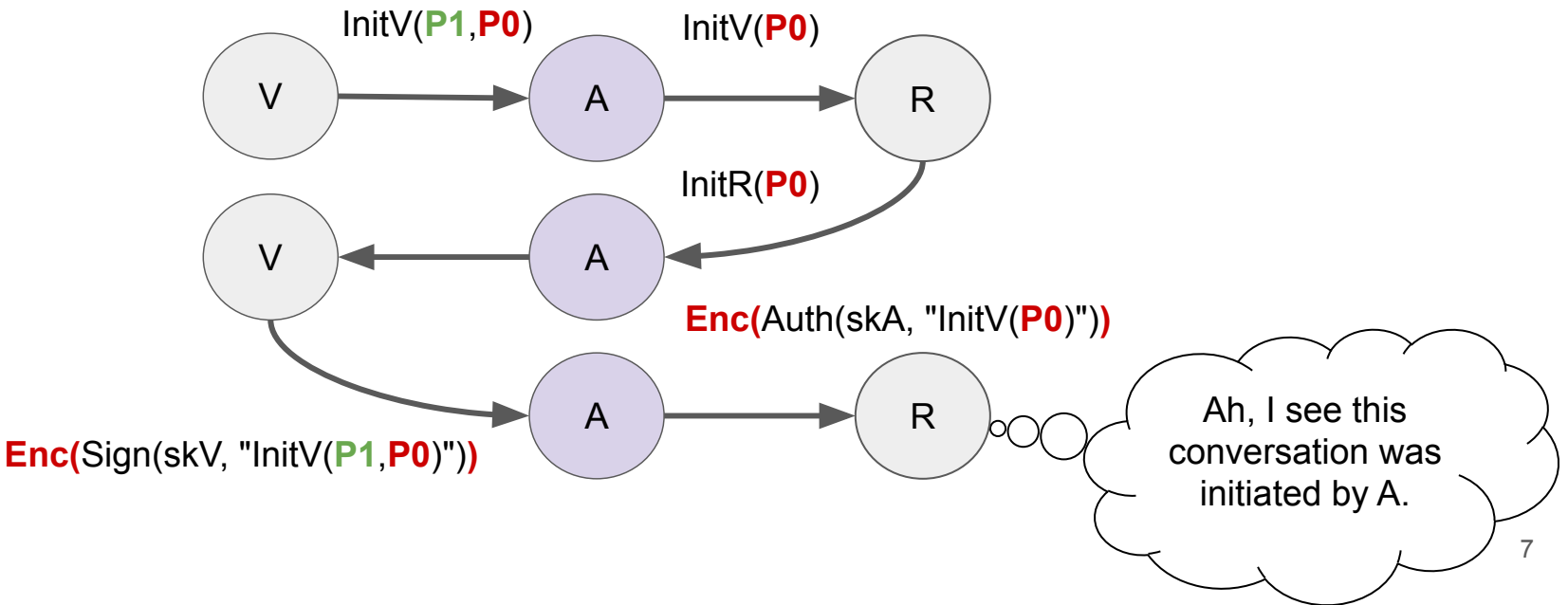
Example attack: Identity misbinding

Step 2: A intercepts forwards R's IKE_INIT_SA message to V as is.



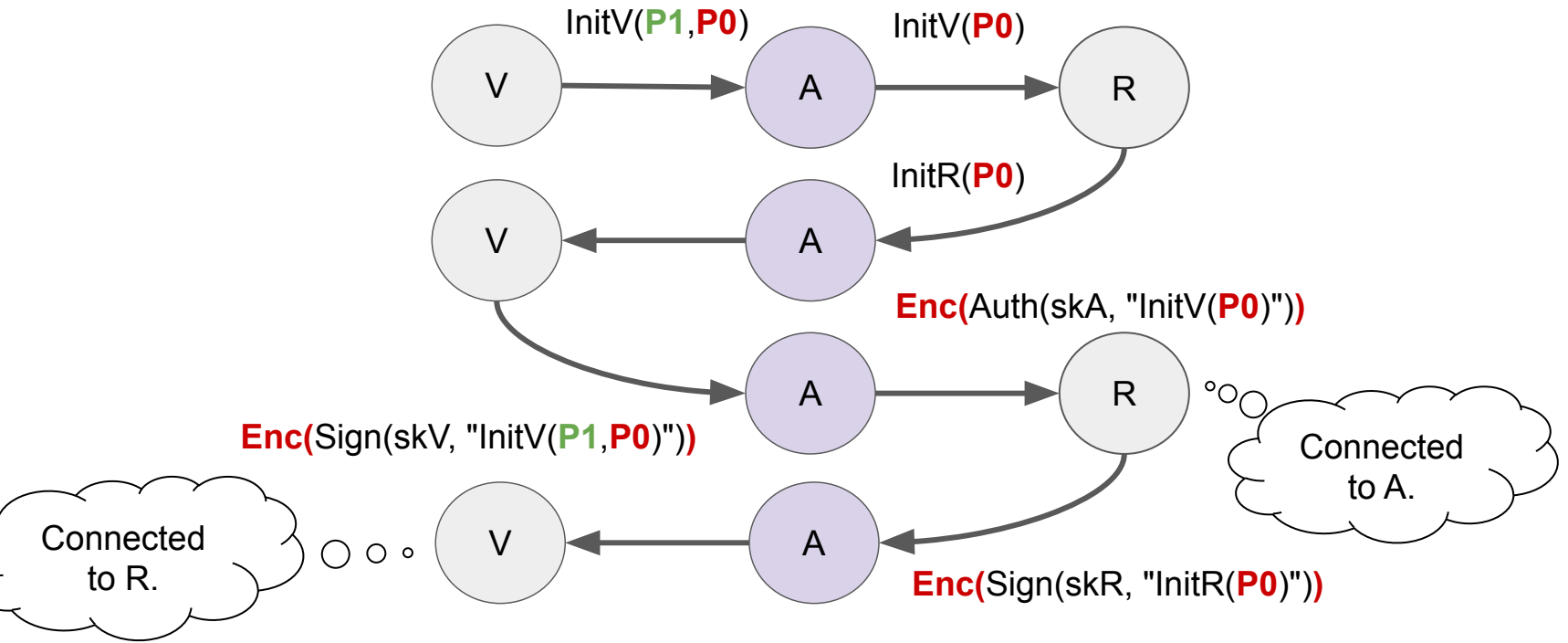
Example attack: Identity misbinding

Step 3: A intercepts V's IKE_AUTH message, decrypts it, replaces V's authentication with its own, re-encrypts, and forwards the modified message to R.



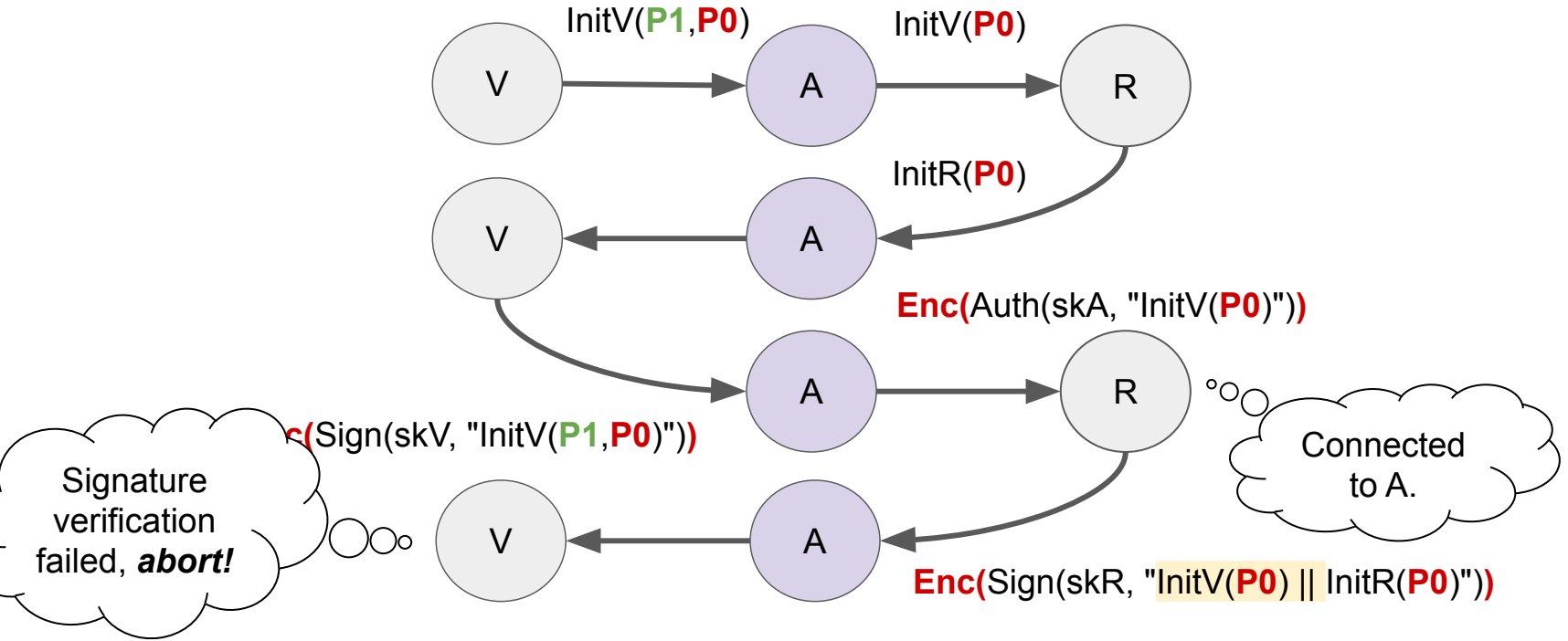
Example attack: Identity misbinding

Step 4: A forward's R's IKE_AUTH message to V as is.



Example attack: Identity misbinding

V and R disagree on the initial IKE_SA_INIT message. Had R confirmed the message it received, then V would have detected the attack.



Downgrade prevention

- Overrides authentication logic: Each party signs ***the entire IKE_SA_INIT exchange*** rather only only its outbound messages (similar to TLS 1.3).
- Support for extension is ***always*** notified in IKE_INIT_SA (peers that don't support the extension will ignore it).
- If you support the extension ***and*** your peer notifies support, then you always use the new authentication logic.

How does this prevent downgrades?

- The parties confirm they've had the same conversation.
- If attacker rewrites IKE_INIT_SA so that it doesn't notify support, then authentication will fail if both support the extension.

Downgrade prevention

When the peer ***doesn't*** notify support:

```
InitiatorSignedOctets =
     RealMessage1 |
     NonceRData | MACedIDForI
```

```
ResponderSignedOctets =
     RealMessage2 |
     NonceIData | MACedIDForR
```

When the peer ***does*** notify support:

```
InitiatorSignedOctets = ZeroPrefix |
     RealMessage2 | RealMessage1 |
     NonceRData | MACedIDForI
```

```
ResponderSignedOctets = ZeroPrefix |
     RealMessage1 | RealMessage2 |
     NonceIData | MACedIDForR
```

Feedback from adoption call

There seems to be consensus the attack works (in theory), and that the proposed mitigation is sufficient and not too difficult to implement. (No public implementation as of yet — but there is a proprietary PoC.) Not everyone agrees the attack is relevant in practice:

- Can be mitigated by requiring strong KE for peers known to support it ← this out-of-bound info is not available in all deployments
- Not relevant until Q day ← downgrade attacks are already relevant today wherever weak KE is supported
- Attack is online, making it hard to pull off ← hard to predict how quickly quantum computers will scale up

Suggested changes:

- Uri Blumenthal, Jun Hu: Mention PPK (RFC 8784) as an alternative mitigation (PR #7)
- Wang Guilin, Chris Patton: Initiator and responder shouldn't sign messages in the same order (PR #8)

Open issues

Issue #10: When resuming a session in which downgrade prevention was negotiated, should we:

1. Renegotiate downgrade prevention? ← Doesn't seem to add much value until there's a fresh key exchange.
2. Indicate negotiation status in the ticket?

Editors think we should do (2.), but that (1.) adds unnecessary complexity. (See PR #11.)

Issue #9: What to call the extension?

- IKE_SA_INIT_AUTH (current)
- IKE_SA_FULL_AUTH
- IKE_SA_DOWNGRADE_PREVENTION

Next steps

- Open source implementation for interop testing
- Codepoint
- Anything else before WGLC?