# PERFECT FORWARD SECRECY FOR EAP-AKA’

The Intercept_

THE GREAT SIM HEIST

How Spies Stole the Keys to the Encryption Castle

OR
HOW THE SPIES ATTACKED
MY PROTOCOL AND I
WANTED TO FIGHT BACK

*Jari Arkko, Karl Norrman, Vesa Torvinen*
*Ericsson Research*

*draft-arkko-eap-aka-pfs-03.txt*[*]

*) Has an IPR notice

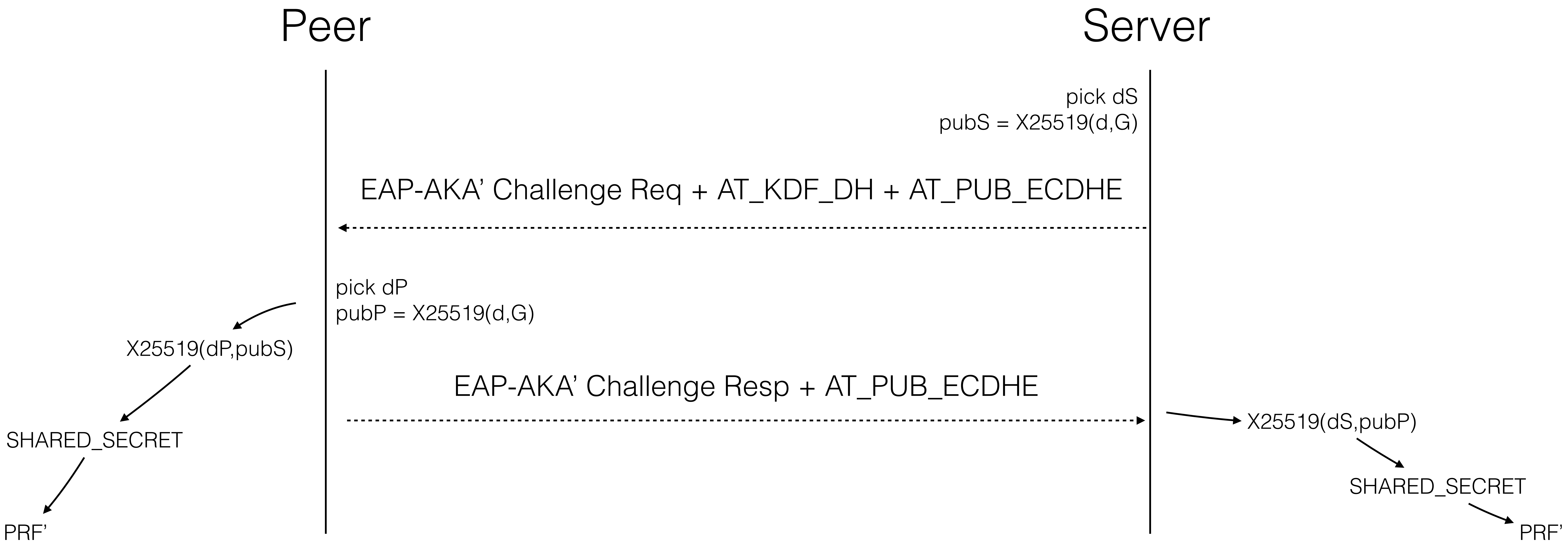
# What’s New

- Took into account reviews
- A more detailed specification
- Renamed attributes and made the protocol use ECDHE and RFC 8031 terminology
- Clarified the use of the negotiation process (only one key sent)
- Clarified the resulting security properties
- Discusses denial-of-service attacks

# The Protocol

- Backwards-compatible extension to add elliptic-curve ephemeral Diffie-Hellman (ECDHE) exchange to EAP-AKA’

- EAP-generated keys provide Perfect Forward Secrecy



Peer

Server

pick dS
pubS = X25519(d,G)

EAP-AKA’ Challenge Req + AT_KDF_DH + AT_PUB_ECDHE

pick dP
pubP = X25519(d,G)

X25519(dP,pubS)

SHARED_SECRET

PRF’

EAP-AKA’ Challenge Resp + AT_PUB_ECDHE

X25519(dS,pubP)

SHARED_SECRET

PRF’

# Simple! Is that it?

There are some tricky parts:

- Backwards compatibility:
  - Avoiding changes to SIM cards or HSS
  - Avoiding extra roundtrips for different cases
- PFS algorithms negotiation process
- Co-existence with other negotiation processes
- Denial-of-Service resistance

# Avoiding Changes to SIM Cards or HSS

- Diffie-Hellman is performed on the phone/authentication server rather than the SIM card and HSS

- This keeps the interfaces to SIM card and to HSS unchanged

- I.e., no changes to credentials or key parts of infrastructure, only EAP implementations

# Avoiding Extra Roundtrips

- Desirable to not have extra roundtrips, when using PFS
- Desirable to not have extra roundtrips if peer turns out not supporting this (when configured to accept downgrade)
- Solution: run both processes in parallel
    - New PFS functionality is just optional attributes — if you do not understand them or want to engage in the new protocol, you can ignore can continue as-is in today's protocol

# PFS Algorithms Negotiation

- AT_KDF_PFS indicates preferred and possible algorithms and groups
  - Currently only X25519
- If preferred algorithm is acceptable, just proceed (no RTT hit)
- Otherwise, peer indicates preference and server accepts in one extra roundtrip
  - Important to avoid MITM editing algorithm lists; AT_MAC protects both offers and responses
  - Only the key for the preferred/selected algorithm is passed in messages, as otherwise we could run out of space

# Co-Existence of PFS and PRF Negotiation

- Both the AT_KDF and AT_KDF_PFS may result in negotiation roundtrips

- Currently defined as separate, i.e., negotiation of each is recognised from a message from the peer carrying only one attribute

- Could probably define this to work in parallel too. Necessary?

# Denial-of-Service Resistance

- The first message comes from the Server; requires the user is an active subscriber. Can be spoofed, but requires attackers use specific, real identities

- There is an order in which computations and checks must occur.

  - When processing the EAP-Request/AKA’-Challenge message the AKA authentication must be checked and succeed before the peer proceeds to calculating or processing the PFS related parameters (S 6.5.4).

  - I.e., parties need to show possession of the long-term secret before heavy calculations.  This limits the Denial-of-Service to specific, authenticated subscribers.

- Further actions (barring a subscriber etc) are possible after this.

# Updated Security Properties

Prevents an attacker who has gained access to the long-term pre-shared secret in a SIM card from being able to decrypt all past communications.

In addition, if the attacker stays merely a passive eavesdropper, the extension prevents attacks against future sessions. This forces attackers to use active attacks instead.

Active attackers with the secrets… all is lost (can be MITM and determine session keys, change negotiations, etc)

# Next Steps

- Feedback?
- WG adoption call on list?