

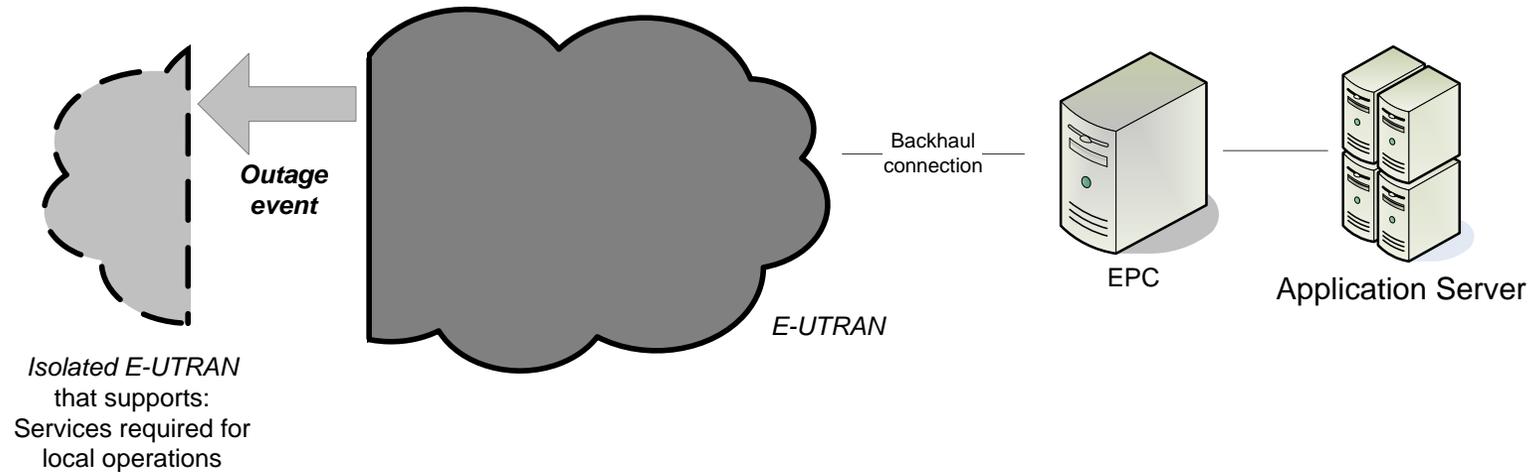
Identity-based authentication and
key agreement for IOPS System-in-a-Box

Intel, S3-15xxxx

Outline

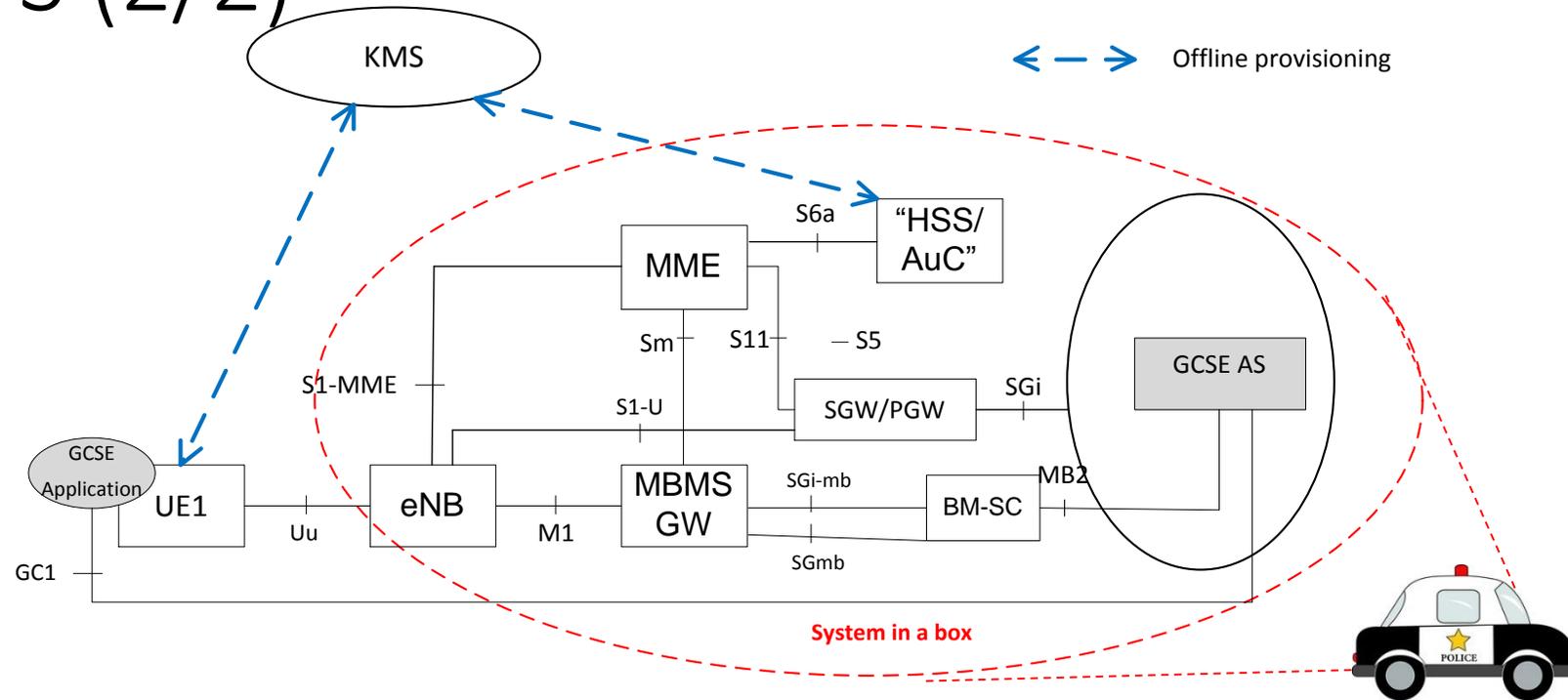
- This contribution discusses authentication and key agreement model for the IOPS (Isolated E-UTRAN Operation for Public Safety) “no backhaul” case
- Focuses on identity-based authentication and key agreement algorithms defined in:
 - RFC 6507 ECCSI for Identity-Based Encryption
 - RFC 6508 SAKKE
 - RFC 6509 MIKEY-SAKKE
- Note that the same algorithms have been adopted in TS 33.303 for ProSe media security
 - Refer to backup slides for correspondence between IETF and TS 33.303 parameter naming

IOPS (1/2)



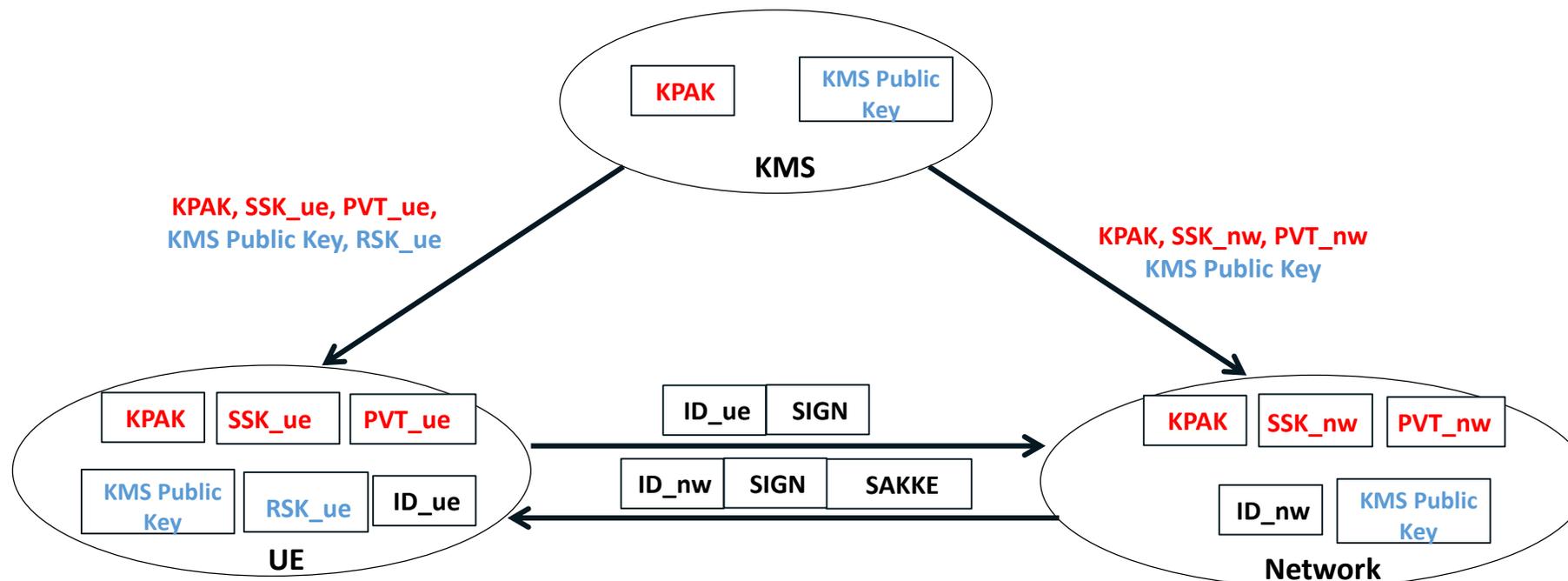
- IOPS (TR 22.346) considers emergency/disaster use cases where the existing E-UTRAN network becomes partly or fully disconnected from the EPC, or where a set of “Nomadic eNBs” (NeNBs) need to be deployed on the fly
- The following scenarios are considered:
 - No backhaul
 - Limited bandwidth signalling only backhaul
 - Limited bandwidth signalling and user data backhaul
- This contribution focuses on the “no backhaul” case. It is assumed that this case is addressed with a System-in-a-Box (see next slide)
- Security credentials other than SIM-card credentials are needed for this case

IOPS (2/2)



- IOPS can be addressed with a “system-in-a-box” network element deployed on the spot
 - It includes rudimentary EPC and App Server functionality, in addition to the eNB(s)
- A Key Management System (KMS) is used to provision credentials for IOPS operation in the UEs and in the “HSS/AuC” function inside the “system-in-a-box”
 - ECCSI (RFC 6507) and SAKKE (RFC 6508) can be used for authentication and key agreement

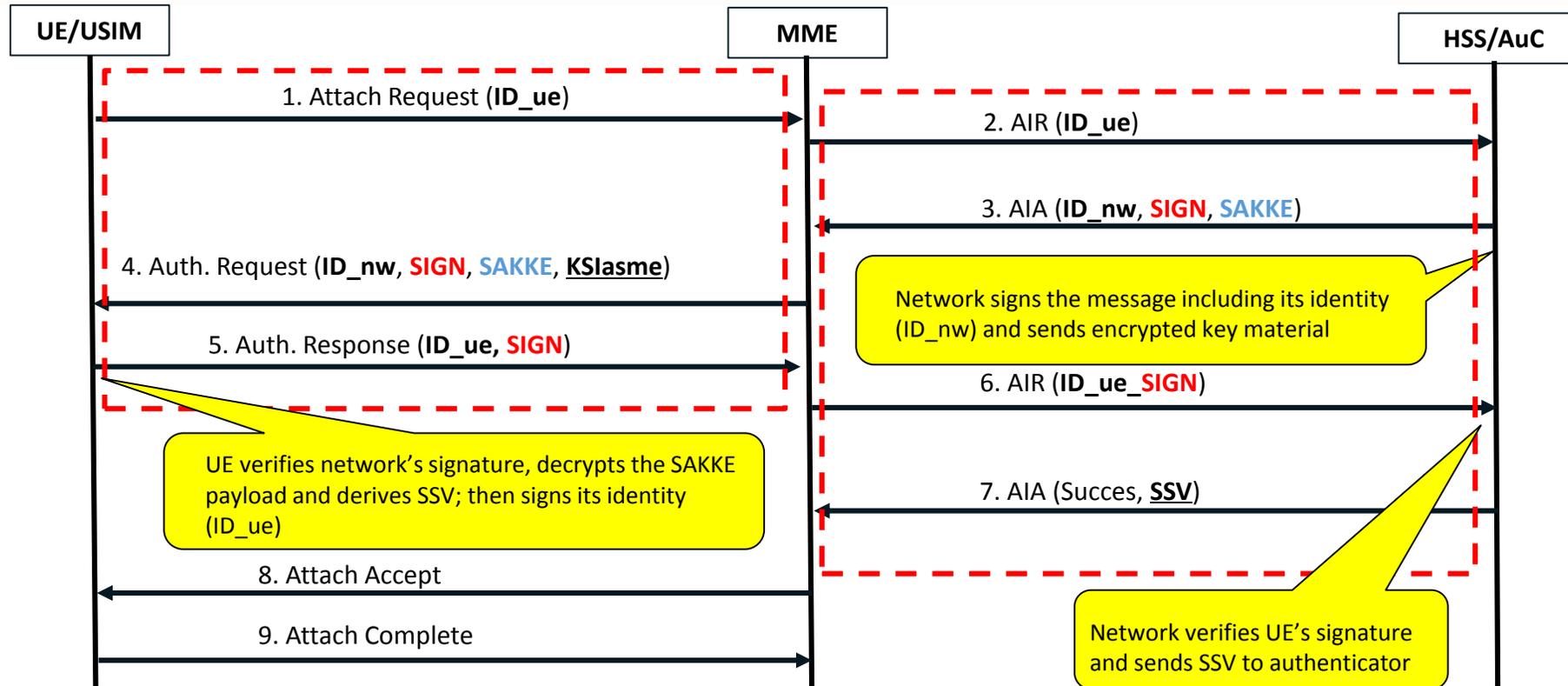
RFC 6507 and RFC 6508 parameters



- Roles for mutual authentication
 - UE (Signer) and Network (Verifier)
 - Network (Signer) and UE (Verifier)
- Roles for secure key distribution
 - Network (Sender) and UE (Receiver)

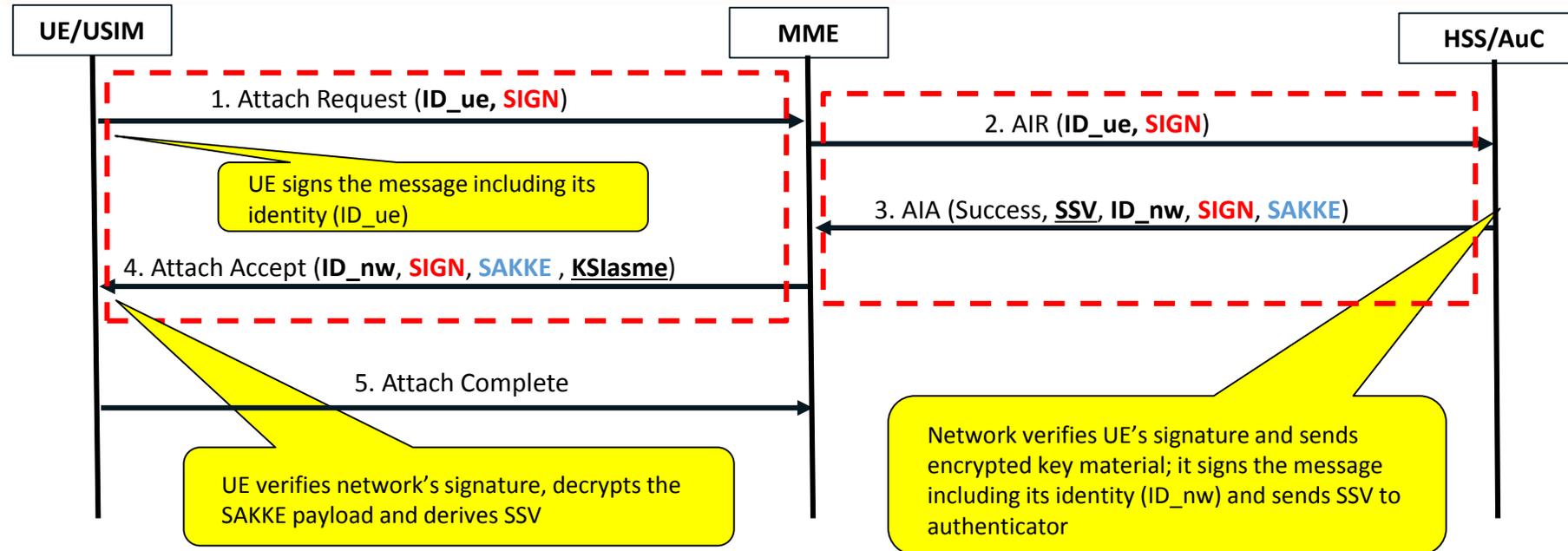
- ID_ue may be an identifier derived from IMSI; it includes a timestamp according to rules in RFC 6509
- ID_nw is a new network identifier that also includes a timestamp. The network can use several different ID_nw identifiers. The ID_nw can be broadcasted in the cell

ID-based authentication and key agreement for IOPS (1/2)



- Possible call flow for authentication and key agreement based on ECCIS/SAKKE
 - Step 3: message is signed (SIGN) using network's identity (ID_nw); the SAKKE payload is encrypted using UE's identity (ID_ue)
 - Step 4: the MME adds KSIasme for key referencing as it does today
 - Step 5: message is signed (SIGN) using UE's identity (ID_ue)

ID-based authentication and key agreement for IOPS (2/2)



- An alternative call flow for authentication and key agreement based on ECCIS/SAKKE
 - Step 1: message is signed (SIGN) using UE's identity (ID_ue)
 - Step 3: message is signed (SIGN) using network's identity (ID_nw); the SAKKE payload is encrypted using UE's identity (ID_ue)
 - Step 4: the MME adds KSIasme for key referencing as it does today

Other

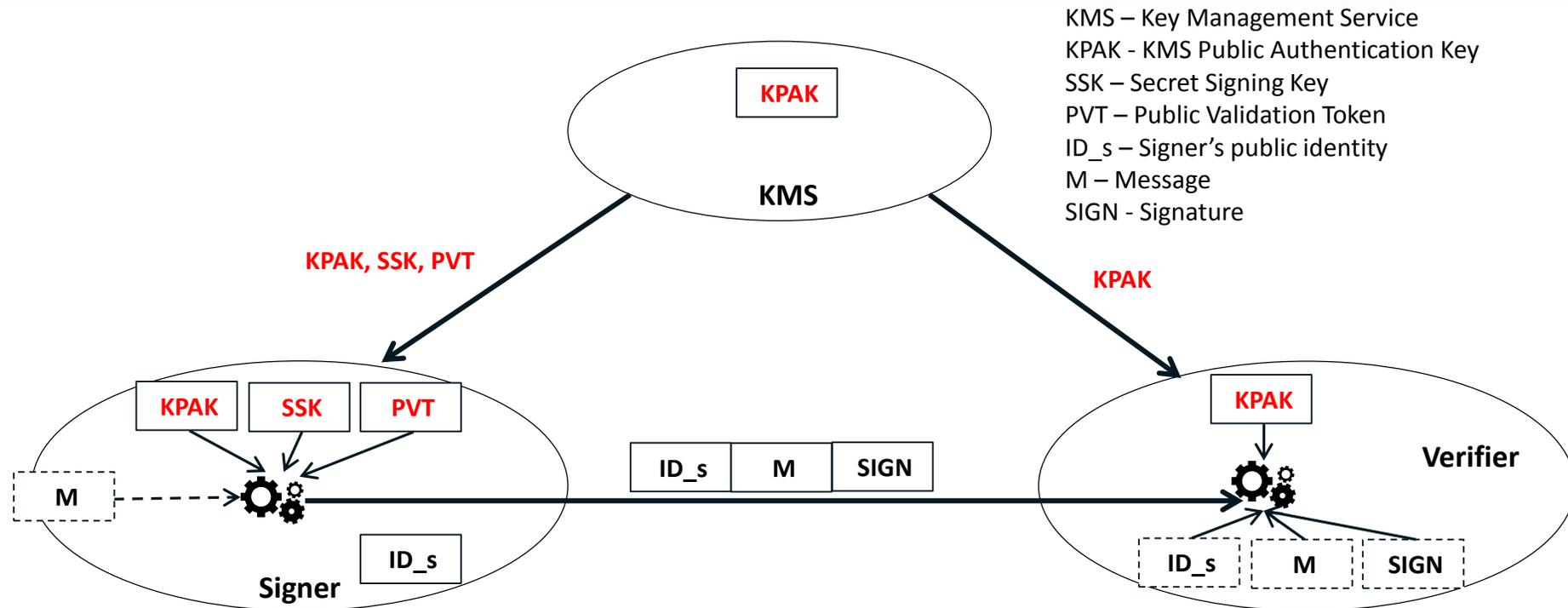
- The Shared Secret Value (SSV) is used to generate a 256-bit KASME
- All other key derivation parameters and mechanisms remain as they are
 - KeNB, KeNB*, NH, NAS count, KSIasme, etc.
- It is FFS whether SIGN and/or SAKKE payload size is compatible with the proposed NAS messages (Attach Request, Attach Accept, Auth. Request, Auth. Response)
- Intel intends to propose a related P-CR and welcomes discussion and cooperation with other companies

Backup

References

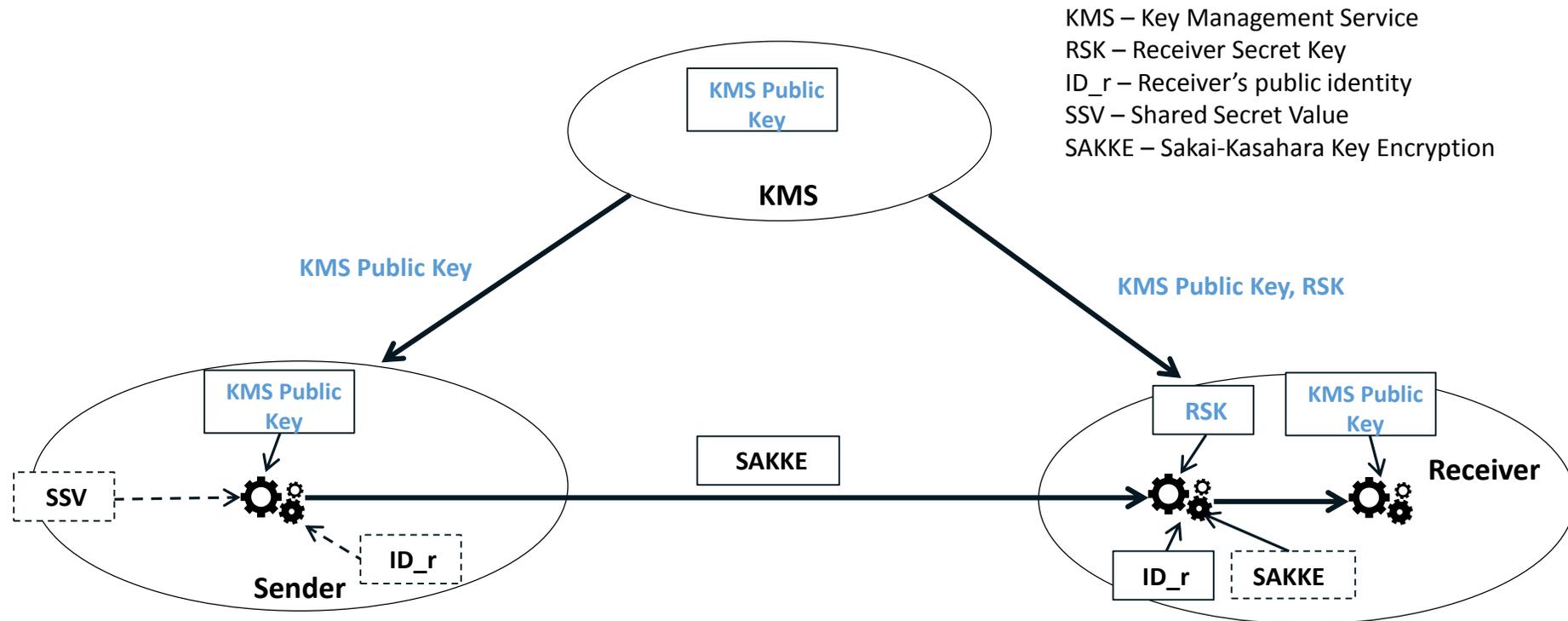
- 3GPP TS 33.303, “ProSe security aspects”
- 3GPP TS 33.401, “3GPP SAE Security architecture”
- IETF RFC 6507, “Elliptic Curve-Based Certificateless Signatures for Identity-Based Encryption (ECCSI)”
- IETF RFC 6508, “Sakai-Kasahara Key Encryption”
- IETF RFC 6509, “MIKEY-SAKKE: Sakai-Kasahara Key Encryption in Multimedia Internet KEYing (MIKEY)”

ECCSI signature for identity authentication (RFC 6507)



- Allows for a message M to be digitally signed by the Signer and then verified by the Verifier
 - Signer uses KPAK, SSK and PVT to sign the message
 - Verifier uses KPAK, ID_s and PVT to verify the signature
- All users are provisioned with the KPAK
- The Signer is provisioned with an (SSK, PVT) pair; PVT is not secret
- PVT is included in the signature (SIGN)

SAKKE for distribution of shared key (RFC 6508)



- Allows the Sender to generate a secret key (SSV) and distribute it safely to the Receiver
 - SSV is encrypted using Receiver's public identity (ID_r) and KMS Public Key
 - The encrypted key (SAKKE payload) is decrypted using RSK and ID_r, and the result is tested using KMS Public Key
- All users are provisioned with the KMS Public Key
- The Receiver is provisioned with RSK

IETF and TS 33.303 parameter naming

| IETF RFC 6507/6508/6509 | TS 33.303 |
|---------------------------------------|---|
| KMS (Key Management Service) | KMS (Key Management System) |
| KPAK (KMS Public Authentication Key) | <i>PubAuthKey</i> (the ECCSI Public Key) |
| SSK (Secret Signing Key) | <i>UserSigningKeySSK</i> (the ECCSI private Key) |
| PVT (Public Validation Token) | <i>UserPubTokenPVT</i> (the ECCSI public validation token) |
| ID_r, ID_s (public identities) | In TS 33.303 today these are UE public identities. For IOPS these are new UE and network public identities |
| SIGN (Signature payload) | "SIGN (ECCSI) payload" |
| RSK (Receiver Secret Key) | <i>UserDecryptKey</i> (the SAKKE "Receiver Secret Key") |
| SSV (Shared Secret Value) | In TS 33.303 today this is the GSK. For IOPS this is equivalent to KASME for IOPS |
| SAKKE (Sakai-Kasahara Key Encryption) | "SAKKE payload" |