Network Working Group
Request for Comments: 4784
Category: Informational

C. Carroll
Ropes & Gray LLP
F. Quick
Qualcomm Inc.

June 2007

Verizon Wireless Dynamic Mobile IP Key Update for cdma2000(R) Networks

Status of This Memo

Copyright Notice

IESG Note

This document describes an existing deployed technology that was developed outside the IETF. It utilizes the RADIUS Access-Reject in order to provision service, which is incompatible with the RADIUS protocol, and practices the sharing of secret keys in public-key cryptosystems, which is not a practice the IETF recommends. The IESG recommends against using this protocol as a basis for solving similar problems in the future.

Abstract

The Verizon Wireless Dynamic Mobile IP Key Update procedure is a mechanism for distributing and updating Mobile IP (MIP) cryptographic keys in cdma2000(R) networks (including High Rate Packet Data, which is often referred to as 1xEV-DO). The Dynamic Mobile IP Key Update (DMU) procedure occurs between the MIP Mobile Node (MN) and RADIUS Authentication, Authorization and Accounting (AAA) Server via a cdma2000(R) Packet Data Serving Node (PDSN) that is acting as a Mobile IP Foreign Agent (FA).

cdma2000(R) is a registered trademark of the Telecommunications Industry Association (TIA).

Table of Contents

1. Introduction

The Verizon Wireless Dynamic Mobile IP Key Update procedure is a mechanism for distributing and updating Mobile IP (MIP) cryptographic keys in cdma2000(R) 1xRTT (1X) [2] and High Rate Packet Data (HRPD) / 1xEV-DO networks [3]. The Dynamic Mobile IP Key Update (DMU) procedure occurs between the Mobile IP Mobile Node (MN) and the home RADIUS [4] (or Diameter [5]) Authentication, Authorization and Accounting (AAA) Server via a cdma2000(R) Packet Data Serving Node (PDSN) that is acting as a Mobile IP Foreign Agent (FA). (In this document, we use the acronym AAAH to indicate the home AAA server as opposed to an AAA server that may be located in a visited system.) This procedure is intended to support wireless systems conforming to Telecommunications Industry Association (TIA) TR-45 Standard IS-835 [6]. DMU, however, could be performed in any MIP network to enable bootstrapping of a shared secret between the Mobile Node (MN) and RADIUS AAA Server.

The DMU procedure utilizes RSA public key cryptography to securely distribute unique MIP keys to potentially millions of cdma2000(R) 1X and HRPD/1xEV-DO Mobile Nodes (MN) using the same RSA public key.

By leveraging the existing cdma2000(R) 1X authentication process, the Dynamic Mobile IP Key Update process employs a mutual authentication mechanism in which device-to-network authentication is facilitated using cdma2000(R) 1X challenge-response authentication, and network-to-device authentication is facilitated using RSA encryption.

By utilizing RSA encryption, the MN (or MN manufacturer) is able to pre-generate MIP keys (and the Challenge Handshake Authentication Protocol (CHAP) key) and pre-encrypt the MIP keys prior to initiation of the DMU procedure. By employing this pre-computation capability, the DMU process requires less computation (by an order of magnitude) during the key exchange than Diffie-Hellman Key Exchange.

1.1. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [1].

2. Basic Dynamic MIP Key Update Mechanism

The DMU procedure is basically an authentication and key distribution protocol that is more easily understood by separately describing the mechanism’s two functional goals: 1) encrypted key distribution and 2) mutual authentication.

2.1. RSA Encrypted Key Distribution

By utilizing RSA public key cryptography, MNs can be pre-loaded with a common RSA public (encryption) key (by the MN manufacturer), while the associated RSA Private (decryption) key is securely distributed from the MN manufacturer to a trusted service provider. Alternatively, a service provider can generate its own RSA public/private key pair and only distribute the RSA public key to MN manufacturers for pre-loading of MNs.

During the manufacturing process, the MN manufacturer pre-loads each MN with the RSA public key. When the MN is powered-up (or client application initiated), the MN can pre-generate and encrypt MIP keys for distribution to the Home RADIUS AAA Server during the DMU process. Alternatively, the MN manufacturer can pre-generate MIP keys, encrypt the MIP key payload, and pre-load the MN with multiple encrypted MIP key payloads to enable the DMU procedure.

During the initial registration process (or when the AAA requires MIP key update), the MN: 1) generates the appropriate MIP keys, CHAP key, and authentication information, 2) uses the embedded RSA public key to encrypt the payload information, 3) and appends the payload to the MIP Registration Request. The Registration Request is sent to the Mobile IP Foreign Agent (FA) via the cellular Base Station (BS) and Packet Data Serving Node (PDSN). When the RADIUS AAA Server receives the encrypted payload (defined later as MIP_Key_Data), the AAA Server uses the RSA Private key to decrypt the payload and recover the MIP keys.

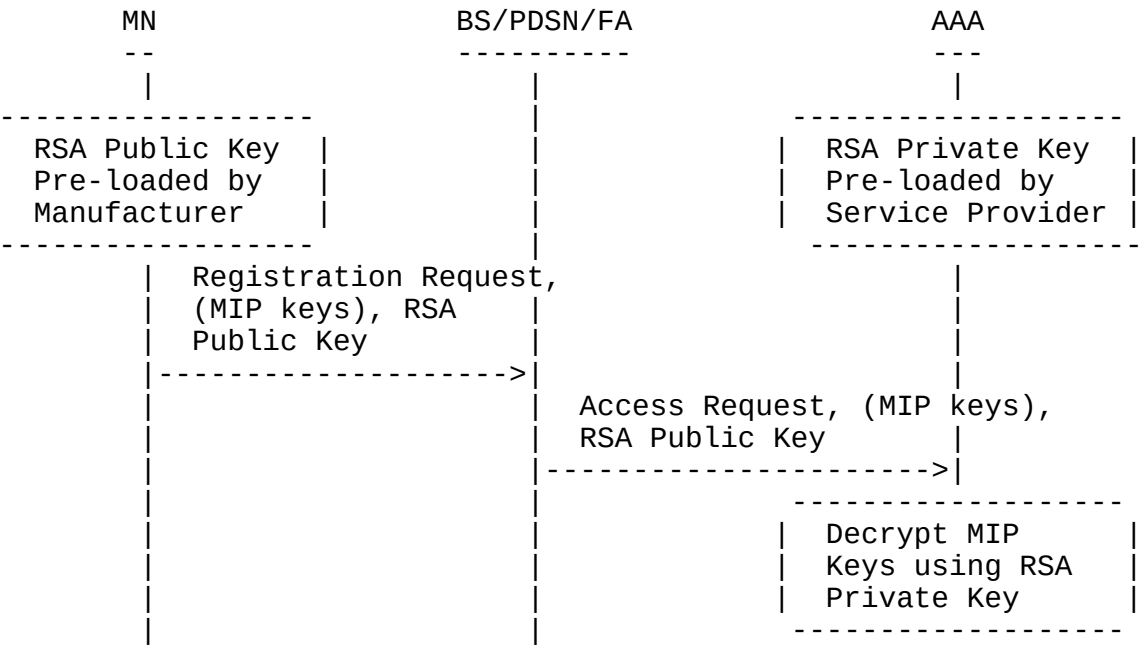

Figure 1. RSA Encrypted Key Distribution

2.2. Mutual Authentication (1X)

Mutual authentication can be achieved by delegation of the MN/device authentication by the RADIUS AAA Server to the cdma2000(R) 1X Home Location Register (HLR) and its associated Authentication Center (AC) [7], while the MN utilizes RSA encryption to authenticate the RADIUS AAA Server.

MN/device authentication via an HLR/AC is based on the assumption that the MN's Mobile Station (MS) has an existing Authentication Key (A-key) and Shared Secret Data (SSD) with the cdma2000(R) 1X network. When MS call origination occurs, the AC authenticates the MS. If authentication is successful, the BS passes the Mobile Station Identifier (MSID) (e.g., Mobile Identification Number (MIN)) to the PDSN. The "Authenticated MSID" is then included in the RADIUS Access Request (ARQ) message [4] sent from the PDSN to the RADIUS AAA server. Because the RADIUS AAA server stores the MSID associated with an MN subscription, the RADIUS AAA server is able to authorize MN access if the "Authenticated MSID" matches the RADIUS AAA MSID, i.e., the RADIUS AAA server is delegating its authentication function to the cdma2000(R) 1X HLR/AC.

RADIUS AAA Server authentication (by the MN) is enabled by including a random number (AAA_Authenticator) in the encrypted payload sent from the MN to the RADIUS AAA Server. Only the possessor of the proper RSA Private key will have the ability to decrypt the payload and recover the unique AAA_Authenticator. If the MN receives the correct AAA_Authenticator (returned by the RADIUS AAA Server), the MN is assured that it is not interacting with a false Base Station (BS).

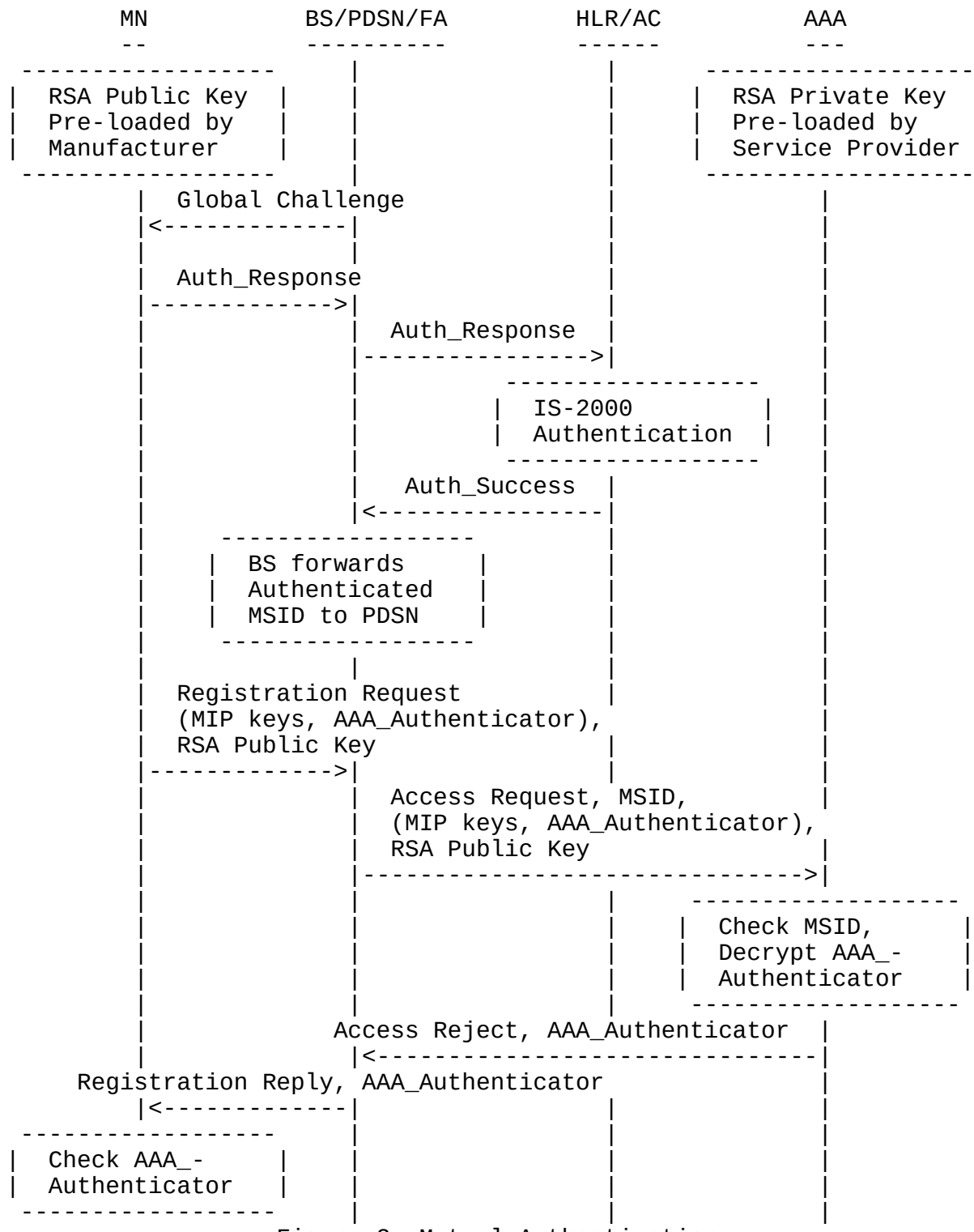

Figure 2. Mutual Authentication

2.3. Encrypted Password Authentication

Because cdma2000(R) A-key/SSD authentication is not available in 1xEV-DO, or a particular cdma2000(R) 1X network may not support A-key authentication, the DMU procedure also includes a random number (MN_Authenticator) generated by the MN (and/or pre-loaded by the manufacturer), which enables the RADIUS AAA Server to optionally authenticate the MN (in 1XEV DO network only).

The MN_Authenticator is transmitted from the MN to the Home AAA Server within the RSA-encrypted MIP_Key_Data payload to prevent interception and possible re-use by an attacker. Ideally, the MN_Authenticator is utilized as a One-Time Password; however, RSA encryption allows the MN_Authenticator to possibly be re-used based on each service provider's key distribution policy.

When the encrypted MIP keys are decrypted at the Home RADIUS AAA Server, the MN_Authenticator is also decrypted and compared with a copy of the MN_Authenticator stored within the Home RADIUS AAA Server. The Home RADIUS AAA Server receives a copy of the MN_Authenticator out-of-band (not using the cdma2000(R) network) utilizing one of numerous possible methods outside the scope of the standard. For example, the MN_Authenticator MAY be: 1) read out by a Point-of-Sale provisioner from the MN, input into the subscriber profile, and delivered, along with the Network Access Identifier (NAI), via the billing/provision system to the Home RADIUS AAA server, 2) verbally communicated to a customer care representative via a call, or 3) input by the user interfacing with an interactive voice recognition server. The out-of-band MN_Authenticator delivery is not specified in this document to maximize the service provider's implementation flexibility.

It is possible for an unscrupulous provisioner or distribution employee to extract the MN_Authenticator prior to the DMU procedure; however, the risk associated with such a disclosure is minimal. Because the HRPD/1xEV-DO MN does not transmit a device identifier during the initial registration process, an attacker, even with a stolen MN_Authenticator, cannot correlate the password with a particular MN device or NAI, which is typically provisioned just prior to DMU procedure initiation.

The MN_Authenticator is typically generated by a random/pseudorandom number generator within the MN. MN_Authenticator generation is initiated by the MN user; however, it may be initially pre-loaded by the manufacturer. When the MN_Authenticator is reset (i.e., a new MN_Authenticator is generated), all MIP_Data_Key payloads using the previous MN_Authenticator are discarded and the MN immediately re-

encrypts a MIP_Key_Data payload containing the new MN_Authenticator. The MN_Authenticator MUST NOT change unless it is explicitly reset by the MN user. Thus, the MN will generate new MIP_Key_Data payloads using the same MN_Authenticator until the MN_Authenticator is updated.

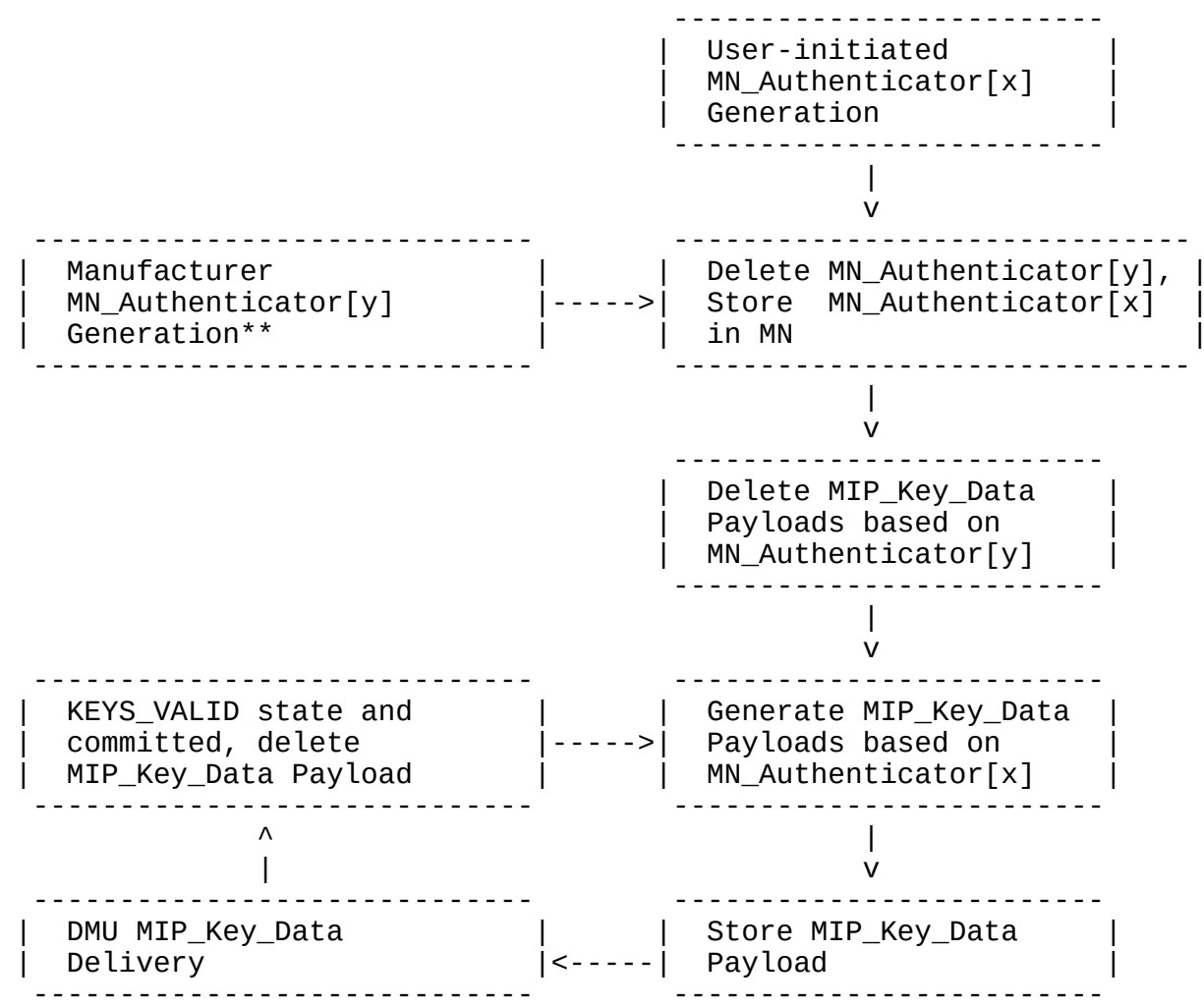

Figure 3. MN_Authenticator and MIP_Key_Data Payload State Machine

**Note: Manufacturer pre-load of MN_Authenticator is not essential since the MN_Authenticator is typically generated by the MN. However, manufacturer pre-load may reduce the provisioner burden of accessing a device such as a modem to recover the MN_Authenticator for entry into the service provider provisioning system.

3. Dynamic MIP Key Update Advantages over OTASP

The DMU procedure has numerous advantages over the current Over-the-Air Service Provisioning (OTASP) [8] procedure, including:

* In DMU, MIP key distribution occurs directly between the MN and AAA Server at the IP Layer. This eliminates the need for an interface between the Over-the-Air Functionality (OTAF) and RADIUS AAA server.

* DMU Supports MIP key distribution for cdma2000(R) 1X and HRPD/1xEV-DO MN. OTASP only supports cdma2000(R) 1X MIP key distribution.

* DMU facilitates MIP key distribution to an MN in a Relay-mode MS. OTASP only delivers the MIP keys to the MS. For example, OTASP cannot deliver MIP keys to a Laptop MN interfacing with an MS modem.

* Pre-encryption of MIP_Key_Data allows the DMU procedure to be an order of magnitude faster than Diffie-Hellman Key Exchange.

* In DMU, an MN manufacturer can pre-generate MIP keys, pre-encrypt the MIP key payload, and pre-load the payload in the MN. Thus, an MN with limited processing power is never required to use RSA encryption. An OTASP device is always forced to perform computationally expensive exponentiations during the key update process.

* In DMU, the MN is protected against Denial-of-Service (DOS) attacks in which a false BS changes the MIP key for MNs in its vicinity. OTASP Diffie-Hellman Key Exchange is vulnerable to a false BS DOS attack.

* DMU utilizes mutual authentication. OTASP Diffie-Hellman Key Exchange does not utilize mutual authentication.

4. Detailed DMU Procedure Description and Requirements

The Verizon Wireless Dynamic Mobile IP Update procedure is a secure, yet extremely efficient mechanism for distributing essential MIP cryptographic keys (e.g., MN-AAAH key and MN-HA key) and the Simple IP CHAP key. The DMU protocol enables pre-computation of the encrypted key material payload, known as MIP_Key_Data. The DMU procedure purposely avoids the use of Public Key Infrastructure (PKI) Certificates, greatly enhancing the procedure’s efficiency.

4.1. RSA Public Key Cryptography

RSA public key encryption and decryption MUST be performed in accordance with RFC 3447 [9] PKCS #1: RSA Encryption Version 1.5. DMU MUST support RSA with a 1024-bit modulus by default. DMU MAY also support 768-bit or 2048-bit RSA, depending on the MN user's efficiency or security requirements. RSA computation speed-ups using a public RSA exponent that is small or has a small number of nonzero bits (e.g., 65537) are acceptable.

4.2. Other Public Key Algorithms

DMU does not preclude the use of other public key technologies. The protocol includes a Public Key Type field that defines the type of encryption used.

4.3. Why No Public Key Infrastructure (PKI)?

DMU is designed to maximize the efficiency of Mobile IP (MIP) key distribution for cdma2000(R) MNs. The use of a public key Certificate would improve the flexibility of the MIP key update process by allowing a Certificate Authority (CA) to vouch for the RSA public key delivered to the MN. Unfortunately, the use of a public key certificate would significantly reduce the efficiency (speed and overhead) of the MIP key update process. For instance, each MN must be pre-loaded with the CA's public key. During the MIP key distribution process, the network must first deliver its RSA public key (in a certificate) to the MN. The MN must then use RSA to decrypt the Certificate's digital signature to verify that the presented RSA public key is legitimate. Such a process significantly increases the number of exchanges, increases air interface overhead, increases the amount of MN computation, and slows the MIP key update process.

Aside from the operational efficiency issues, there are numerous policy and procedural issues that have previously hampered the deployment of PKI in commercial networks.

On a more theoretical basis, PKI is likely unnecessary for this key distribution model. PKI is ideal for a Many-to-Many communications model, such as within the Internet, where many different users interface with many different Websites. However, in the cellular/PCS Packet Data environment, a Many-to-One (or few) distribution model exists, in which many users interface with one wireless Carrier to establish their Mobile IP security associations (i.e., cryptographic keys).

4.4. Cryptographic Key Generation

The DMU procedure relies on each MN to randomly/pseudo-randomly generate the MN_AAAH key, MN_HA key, and Simple IP CHAP key. Each MN MUST have the capability to generate random/pseudo-random numbers in accordance with the guidelines specified in RFC 4086 "Randomness Requirements for Security".

Although it may be more secure for the network to generate cryptographic keys at the RADIUS AAA server, client cryptographic key generation is acceptable due to the significant efficiency improvement in the update process via pre-generation and pre-encryption of the MIP keys.

4.5. MIP_Key_Data Payload

MIP cryptographic keys (MN_AAAH key and MN_HA key) and the Simple IP CHAP key are encapsulated and encrypted into a MIP_Key_Data Payload (along with the AAA_Authenticator and MN_Authenticator). The MIP_Key_Data Payload is appended to the MN's MIP Registration Request (RRQ) as a MIP Vendor/Organization-Specific Extension (VSE) (see RFC 3115 [10] Mobile IP Vendor/Organization-Specific Extensions). When the PDSN converts the MIP RRQ to a RADIUS Access Request (ARQ) message, the MIP_Key_Data Payload is converted from a MIP Vendor/Organization-Specific Extension to a Vendor Specific RADIUS Attribute (VSA).

Upon receipt of the RADIUS Access Request, the RADIUS AAA Server decrypts the MIP_Key_Data payload using the RSA private (decryption) key associated with the RSA public (encryption) used to encrypt the MIP_Key_Data payload. The MIP_Key_Data is defined as follows:

MIP_Key_Data = RSA_Public_Key [MN_AAAH key, MN_HA key, CHAP_key, MN_Authenticator, AAA_Authenticator], Public_Key_ID, DMUV

Where:

MN_AAAH key = 128-bit random MN / RADIUS AAA Server key (encrypted)

MN_HA key = 128-bit random MN / Home Agent (HA) key (encrypted)

CHAP_key = 128-bit random Simple IP authentication key (encrypted)
Note: the Simple IP CHAP key is not the same as the AT-CHAP key used for A12 Interface authentication [11].

MN_Authenticator = 24-bit random number (displayed as an 8 decimal digit number). (To be used for 1xEV-DO networks.) (encrypted)

AAA_Authenticator = 64-bit random number used by MN to authenticate the RADIUS AAA Server. (encrypted)

DMU Version (DMUV) = 4-bit identifier of DMU version.

Public Key Identifier (Public_Key_ID) = PKOID, PKOI, PK_Expansion, ATV

Where:

Public Key Organization Identifier (PKOID) = 8-bit serial number identifier of Public Key Organization (PKO) that created the Public Key.

Public Key Organization Index (PKOI) = 8-bit serial number used at PKO discretion to distinguish different public/private key pairs.

PK_Expansion = 8-bit field to enable possible expansion of PKOID or PKOI fields. (Note: Default value = 0xFF)

Algorithm Type and Version (ATV) = 4-bit identifier of the algorithm used.

Note: If 1024-bit RSA is used, the encrypted portion of the payload is 1024 bits (128 bytes) long. With the 28-bit Public Key Identifier and 4-bit DMUV, the total MIP_Key_Data payload is 132 bytes long.

4.6. RSA Key Management

The wireless service provider or carrier MUST generate the RSA Public/Private key pair(s). An organization within the service provider MUST be designated by the service provider to generate, manage, protect, and distribute RSA Private keys (to the RADIUS AAA Server) and public keys (to the MN manufacturers) in support of the DMU procedure.

Each RSA public/private key pair, generated by the wireless carrier, MUST be assigned a unique Public Key Identifier in accordance with Section 9.

RSA Private keys MUST be protected from disclosure to unauthorized parties. The service provider organization with the responsibility of generating the RSA public/private key pairs MUST establish an RSA key management policy to protect the RSA Private (decryption) keys.

RSA public keys MAY be freely distributed to all MN manufacturers (along with the Public Key Identifier). Because one RSA public key

can be distributed to million of MNs, it is acceptable to distribute the RSA public key (and Public Key Identifier) to MN manufacturers via e-mail, floppy disk, or a Website. The preferred method is to simply publish the RSA public key and associated Public Key Identifier in the DMU Requirements document sent to each MN manufacturer/OEM.

When public keys are distributed, the public keys MUST be protected against alteration. If an invalid public key is programmed into a terminal, the terminal may be denied service because DMU cannot be performed successfully.

RSA Private keys MAY be loaded into the RADIUS AAA server manually. Access to the RADIUS AAA Server RSA Private keys MUST be restricted to authorized personnel only.

The wireless service provider MAY accept RSA Private key(s) (and Public Key Identifier) from MN manufacturers that have preloaded MNs with manufacturer-generated RSA public keys.

4.7. RADIUS AAA Server

The RADIUS AAA Server used for DMU MUST support the DMU Procedure. The AAA Server MUST support RSA public key cryptography and maintain a database of RSA Private (decryption) keys indexed by the Public Key Identifier.

Delivery of the RSA Private key(s) to an AAA Server from the MN manufacturer(s) is outside the scope of this document. However, RSA Private key(s) delivery via encrypted e-mail or physical (mail) delivery is likely acceptable.

Access to the RADIUS AAA Server MUST be limited to authorized personnel only.

The RADIUS AAA Server MUST support 1024-bit RSA decryption.

The RADIUS AAA Server MUST maintain a database of RSA public/private key pair indexed by the Public Key Identifier.

The RADIUS AAA Server MUST support the RADIUS attributes specified in Section 8.

The RADIUS AAA Server MUST support a subscriber-specific MIP Update State Field. When the MIP Update State Field is set to UPDATE KEYS (1), the RADIUS AAA Server MUST initiate the DMU procedure by including the MIP_Key_Request attribute in an Access Reject message sent to the PDSN. The MIP Update State Field MAY be set to UPDATE

KEYS (1) by the service provider's Billing/Provisioning system based on IT policy. Upon verification of MN-AAA Authentication Extension using the decrypted MN_AAA key, the RADIUS AAA Server MUST set the MIP Update State Field to KEYS UPDATED (2). Upon verification of the MN-Authentication Extension on a subsequent RRQ/ARQ, the RADIUS AAA Server MUST set the MIP Update State Field to KEYS VALID (0).

Note that the inclusion of a vendor-specific attribute in the Access Reject message is not consistent with Section 5.44 of [4]. A RADIUS AAA server that supports DMU SHOULD NOT include a vendor-specific attribute if the corresponding Access Request message was not received from a DMU-compliant PDSN. This use of Access Reject is strongly discouraged for any future work based on this document. Future work should consider the use of Access-Challenge to carry this vendor-specific attribute.

The RADIUS AAA Server MUST maintain a MIP Update State Field, for each subscription, in one of three states (0 = KEYS VALID, 1 = UPDATE KEYS, 2 = KEYS UPDATED).

The RADIUS AAA Server MUST decrypt the encrypted portion of the MIP_Key_Data payload using the appropriate RSA Private (decryption) key.

The RADIUS AAA Server MUST check the MN_AAA Authentication Extension of the DMU RRQ using the decrypted MN_AAA key.

The RADIUS AAA Server MUST include the AAA_Authenticator in the Access Accept as a Vendor-Specific RADIUS Attribute.

The RADIUS AAA Server MUST support the MN_Authenticator options specified in Section 6.1.

The RADIUS AAA Server MUST comply with DMU Procedure failure operation specified in Section 5.

The RADIUS AAA Server MUST support manual hexadecimal entry of MN_AAA key, MN_HA key, and Simple IP CHAP key via the AAA GUI for each subscription.

The RADIUS AAA Server MUST provide a mechanism to validate the MIN/International Mobile Subscriber Identity (IMSI). When the MIN/IMSI validation is on, the RADIUS AAA Server MUST compare the MIN/IMSI sent from the PDSN with the MIN/IMSI in the AAA subscription record/profile. If the MINs or IMSIs do not match, the RADIUS AAA Server MUST send an Access Reject to the PDSN/FA. The Access Reject MUST NOT contain a MIP Key Data request

When the "Ignore MN_Authenticator" bit is not set, the RADIUS AAA Server MUST check whether MN_AuthenticatorMN = MN_AuthenticatorAAA. If the MN_Authenticators do not match, the RADIUS AAA Server MUST send an Access Reject to the PDSN/FA. The Access Reject MUST NOT contain a MIP_Key_Data request.

The RADIUS AAA Server MUST include its PKOID (or another designated PKOID) in the MIP_Key_Request RADIUS Attribute.

The RADIUS AAA Server MUST compare the PKOID sent in the MIP_Key_Data RADIUS Attribute with a list of valid PKOIDs in the RADIUS AAA Server. If the PKOID is not valid, the RADIUS AAA Server MUST send an Access Reject to the PDSN with the "Invalid Public Key" Verizon Wireless RADIUS Vendor Specific Attribute (VSA). Note: the same RADIUS attribute may be assigned a different Vendor identifier.

Note that the inclusion of a vendor-specific attribute in the Access Reject message is not consistent with section 5.44 of [4]. A RADIUS AAA server that supports DMU SHOULD NOT include a vendor-specific attribute if the corresponding Access Request message was not received from a DMU-compliant PDSN. This use of Access Reject is strongly discouraged for any future work based on this document. Future work should consider the use of Access-Challenge to carry this vendor-specific attribute.

The RADIUS AAA Server MUST support delivery of the MN-HA key using 3GPP2 RADIUS VSAs as specified in 3GPP2 X.S0011-005-C. The 3GPP2 VSAs used are the MN-HA Shared Key (Vendor-Type = 58) and MN-HA Security Parameter Index (SPI) (Vendor-Type = 57).

The RADIUS AAA Server SHOULD always accept an Access Request from a cdma2000(R) Access Node (AN) for a particular subscriber when the UPDATE KEYS (1) and KEYS UPDATED (2) states are set. In the KEYS VALID (0) state, the RADIUS AAA Server MUST check the Access Request normally.

The RADIUS AAA Server MUST reject an Access Request with the MIP_Key_Data RADIUS Attribute while the RADIUS AAA Server is in the KEYS VALID state, i.e., the AAA MUST NOT allow an unsolicited key update to occur.

4.8. MN (Handset or Modem)

The MN manufacturer MUST pre-load the Wireless Carrier RSA public key (and Public Key Identifier).

The MN manufacturer MUST pre-generate and pre-load the MN_Authenticator.

The MN MUST support 1024-bit RSA Encryption using the pre-loaded RSA public key.

The MN MUST support MN_AAA, MN_HA, and CHAP random/pseudo-random key generation (in accordance with RFC 4086).

The MN MUST support random/pseudo-random AAA_Authenticator and MN_Authenticator generation (in accordance with RFC 4086).

Upon power-up of an MN handset or launch of the MN client, the MN MUST check whether a MIP_Key_Data payload has been computed. If no MIP_Key_Data payload exists, the MN MUST generate and store a MIP_Key_Data payload. The MN MUST maintain at least one pre-generated MIP_Key_Data payload.

The MN MUST construct the MIP_Key_Data payload in accordance with Section 4.5.

The MN MUST initiate the DMU Procedure upon receipt of a MIP Registration Reply (RRP) with the MIP_Key_Request Verizon Wireless Vendor/Organization-Specific Extension (VSE).

Upon receipt of an RRP including the MIP_Key_Request, the MN MUST check the PKOID sent in the MIP_Key_Request. If the MN has a public key associated with the PKOID, the MN MUST encrypt the MIP_Key_Data payload using that public key.

The MN MUST have the capability to designate one public key as the default public key if the MN supports multiple public keys.

The MN MUST insert the Verizon Wireless MIP_Key_Data VSE (or another Organization-specific MIP_Key_Data VSE) after the Mobile-Home Authentication Extension, but before the MN-AAA Authentication Extension. The MIP_Key_Data Extension must also be located after the FA Challenge Extension, if present.

Note: The order of the extensions is important for interoperability. After the FA receives the Access Accept from the RADIUS AAA server, the FA may strip away all MIP extensions after the Mobile-Home Authenticator. If this occurs, it is not necessary for the HA to process the DMU extensions. Other compatibility problems have also been identified during testing with FAs from various vendors who place extensions in various locations. Explicit placement of the extensions eliminates these issues.

Upon initiation of the DMU Procedure, the MN MUST compute the MIP authentication extensions using the newly-generated temporary MN_AAA and MN_HA keys. Upon receipt of the AAA_Authenticator MIP Extension,

the MN MUST compare the AAA_AuthenticatorMN (sent in the encrypted MIP_Key_Data payload) with the AAA_AuthenticatorAAA (returned by the RADIUS AAA Server). If both values are the same, the MN MUST designate the temporary MN_AAA, MN_HA key, and the Simple IP CHAP key as permanent. The MN MUST set its MIP Update State field to KEYS VALID.

The MN MUST support reset (re-generation) of the MN_Authenticator by the MN user as specified in Section 6.2.

The MN MUST enable the MN user to view the MN_Authenticator. MN_Authenticator (24-bit random number) MUST be displayed as an 8 decimal digit number as specified in Section 6.2.

The MN manufacturer MUST pre-load each MN with a unique random 24-bit MN_Authenticator.

Upon reset of the MN_Authenticator, the MN MUST delete all MIP_Key_Data payloads based on the old MN_Authenticator and generate all subsequent MIP_Key_Data payloads using the new MN_Authenticator (until the MN_Authenticator is explicitly re-set again by the MN user).

The MN MUST support manual entry of all cryptographic keys such as the MN_AAA, MN_HA, and Simple IP CHAP key. MN MUST support hexadecimal digit entry of a 128-bit key. (Note: certain Simple IP devices only enable ASCII entry of a password as the CHAP key. It is acceptable for future devices to provide both capabilities, i.e., ASCII for a password or hexadecimal for a key. The authors recommend the use of strong cryptographic keys.)

The MN MUST support the Verizon Wireless MIP Vendor/Organization-Specific Extensions specified in Section 9.

The MN MUST update the RRQ Identification field when re-transmitting the same MIP_Key_Data in a new RRQ.

The MN MUST comply with the DMU Procedure failure operation specified in Section 5.

The RSA public key MAY be stored in the MN flash memory as a constant while being updatable via software patch.

4.9. PDSN / Foreign Agent (FA)

The PDSN MUST support the Verizon Wireless RADIUS Vendor-Specific Attributes (VSA) specified in Section 8 and the Verizon Wireless MIP Vendor/Organization-Specific Extensions (VSEs) specified in Section 9.

The PDSN MAY support the RADIUS VSAs specified in Section 8 and the MIP VSEs specified in Section 9 using another Organization identifier.

Upon receipt of an Access Reject containing the MIP_Key_Update_Request VSA, PDSN MUST send an RRP to the MN with the MIP_Key_Request VSE. The PDSN MUST use the RRP error code = 89 (Vendor Specific) and MUST not tear down the PPP session after transmission.

Upon receipt of an Access Reject containing the AAA_Authenticator VSA, the PDSN MUST send an RRP with the AAA_Authenticator MIP VSE. The PDSN MUST use the RRP error code = 89 (Vendor Specific) and MUST NOT tear down the PPP session after transmission.

Upon receipt of an Access Reject containing the Public Key Invalid VSA, the PDSN MUST send an RRP with the Public Key Invalid MIP VSE. The PDSN MUST use the RRP error code = 89 (Vendor Specific) and MUST NOT tear down the PPP session after transmission.

Note that the inclusion of a vendor-specific attribute in the Access Reject message is not consistent with section 5.44 of [4]. A PDSN that supports DMU MUST accept an Access Reject message containing a vendor-specific attribute. This use of Access Reject is strongly discouraged for any future work based on this document. Future work should consider the use of Access-Challenge to carry this vendor-specific attribute.

Upon receipt of an RRQ with the MIP_Key_Data VSE, the PDSN MUST convert the RRQ to an ARQ with the MIP_Key_Data VSA. The PDSN MUST send the ARQ to the RADIUS AAA server.

The PDSN/FA MUST comply with the DMU Procedure failure operation specified in Section 5.

The PDSN/FA MUST include the PKOID from the Access Reject MIP_Key_Update_Request VSA in the MIP_Key_Request MIP VSE sent to the MN.

4.10. Home Agent (HA)

The HA MUST support the Verizon Wireless MIP Vendor/Organization-Specific Extensions (VSEs) specified in Section 9. (Note: the HA may not encounter a DMU MIP extension if the FA strips away all extensions after the Mobile-Home authentication extension.)

The HA MAY support the MIP VSEs specified in Section 9 using another Organization identifier. (Note: the HA may not encounter a DMU MIP extension if the FA strips away all extensions after the Mobile-Home authentication extension.)

The HA MUST support delivery of the MN-HA key from the Home RADIUS AAA server using 3GPP2 RADIUS Vendor-Specific Attributes (VSA) as specified in 3GPP2 X.S0011-005-C. The 3GPP2 VSAs used are the MN-HA Shared Key (Vendor-Type = 58) and the MN-HA SPI (Vendor-Type = 57).

4.11. DMU Procedure Network Flow

This section provides a flow diagram and detailed description of the process flow involving the Dynamic Mobile IP Update procedure process within the IS-2000 network.

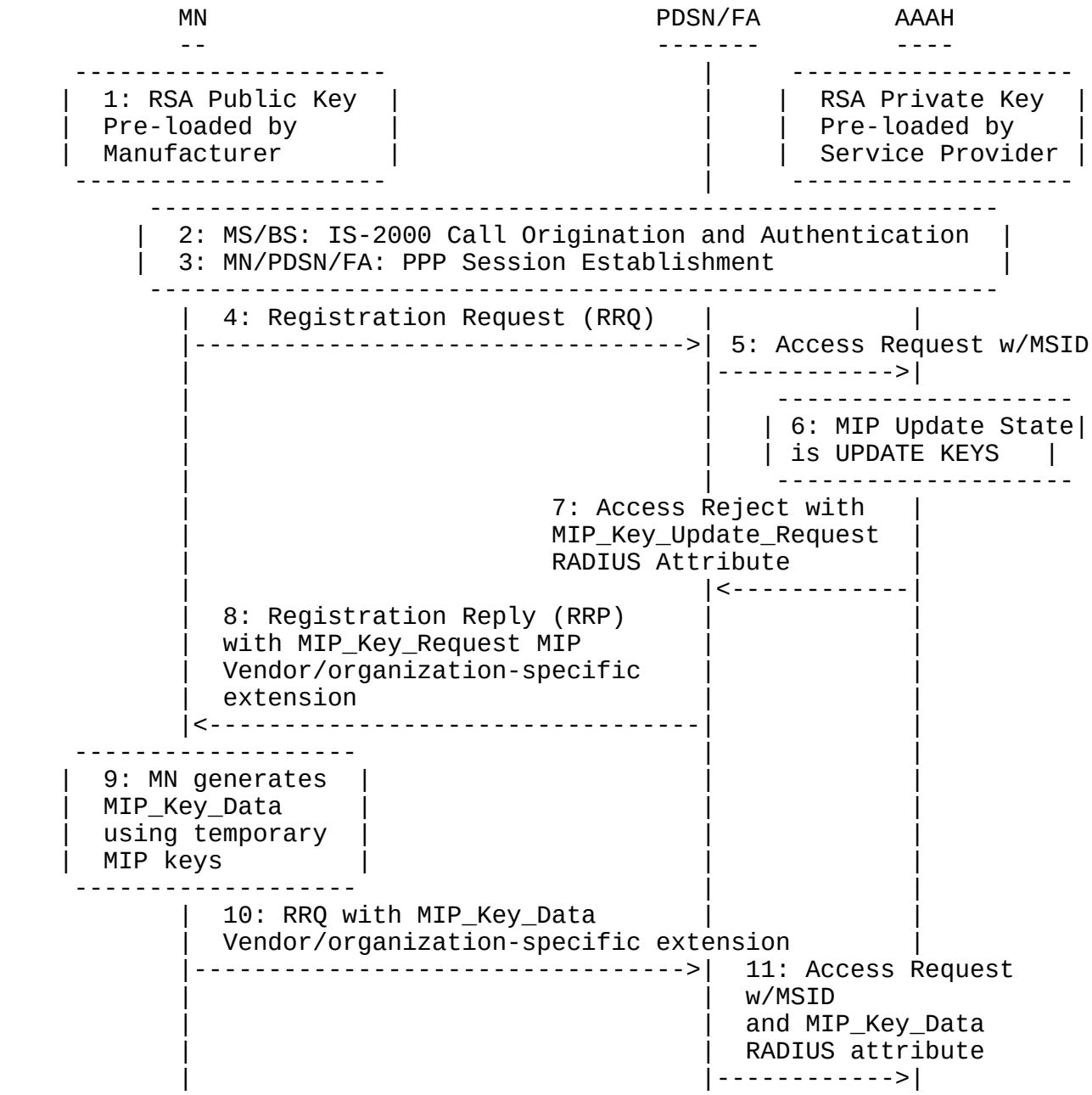

Figure 4. DMU Procedure Flow (part 1)

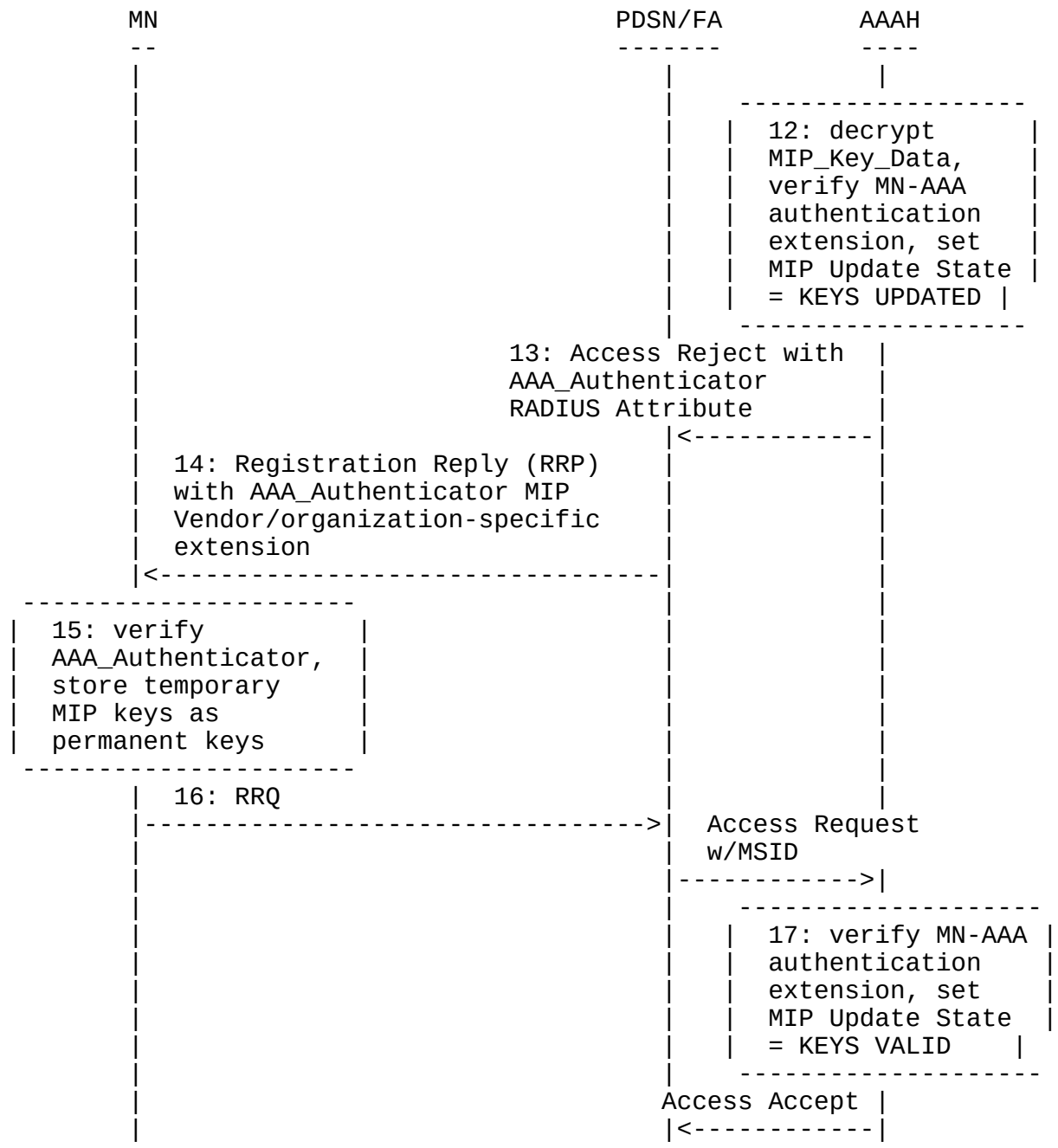

Figure 4. DMU Procedure Flow (part 2)

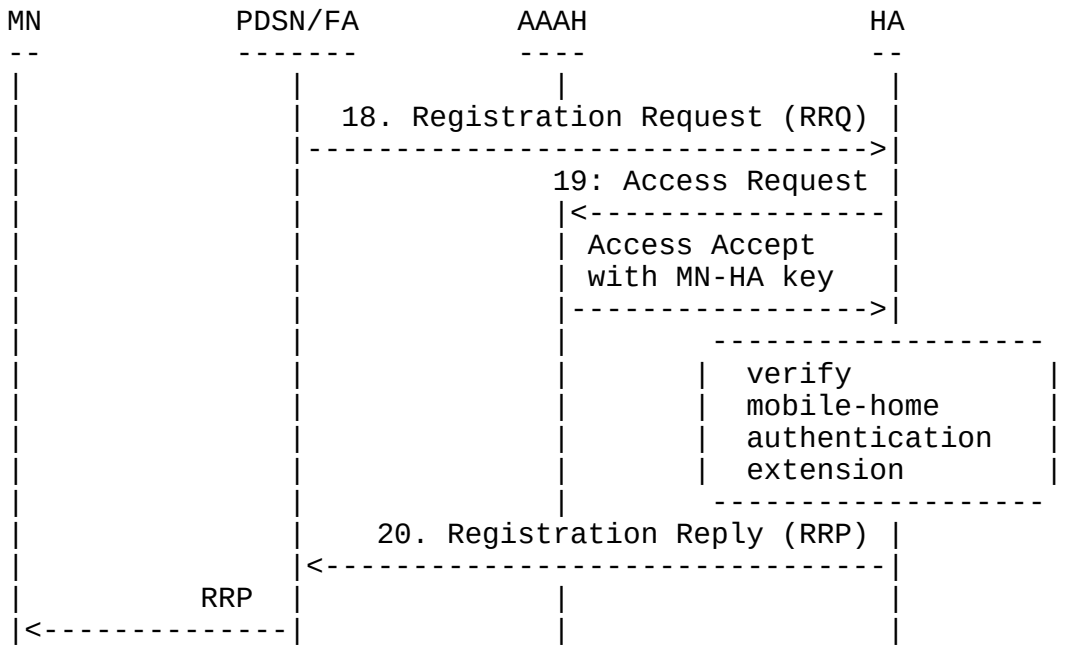

Figure 4. DMU Procedure Flow (part 3)

Each step in the Figure 4 DMU Process is described as follows:

1. Each RSA public/private key pair MUST be generated in accordance with RFC 3447. Each public/private key pair MUST be assigned a unique Public Key Identifier (PKOID) by its creator.

   If the service provider does not generate the public/private key pair and deliver the RSA public key to the MN manufacturer for pre-installation in the MN, the MN manufacturer MUST generate the RSA public/private key pair (using a 1024-bit modulus) and pre-load all MNs with the RSA public (encryption) key. The MN manufacturer MUST distribute the RSA Private (decryption) key, in a secure manner, to the appropriate service provider.

2. Assuming that the cdma2000(R) 1X MN has been provisioned with an A-key and SSD, the cdma2000(R) 1X MS initiates a call origination and authenticates itself to the IS-2000 network. Upon IS-2000 authentication success, the BS sends the "authenticated" MSID (e.g., MIN) to the PDSN.

3. The MN and PDSN establish a PPP session.

4. The MN sends a MIP Registration Request (RRQ) to the PDSN.

5. The PDSN converts the MIP RRQ into a RADIUS Access Request (ARQ) message, includes the MSID in the ARQ, and forwards the ARQ to the Home RADIUS AAA server.

6. The RADIUS AAA Server compares the authenticated MSID (sent from the PDSN) with the MSID in its subscriber database (associated with the NAI). If the AAA MIP Update State Field is set to UPDATE KEYS (1), the RADIUS AAA Server rejects Packet Data access and orders a MIP key update.

7. The RADIUS AAA Server sends an Access Reject (code = 3) message to the PDSN with the MIP_Key_Update_Request RADIUS VSA.

8. The PDSN converts the Access Reject to a MIP Registration Reply (RRP) with a MIP_Key_Request MIP VSE and sends the RRP to the MN. RRP Code = 89 (Vendor Specific).

9. The MN sets the MN MIP Update State = UPDATE KEYS. If the MN has no pre-generated and pre-encrypted MIP_Key_Data payload, the MN MUST generate the MN_AAA key, MN_HA key, Chap key, MN_Authenticator, and AAA_Authenticator in accordance with RFC 4086. Except for the Public Key Identifier, all generated values MUST be encrypted using the pre-loaded RSA public (encryption) key. The newly generated MN_AAATEMP Key and MN_HATEMP MUST be used to calculate the MN-AAA and Mobile-Home Authentication Extensions for the current RRQ. Note: the MN MAY pre-compute the MIP_Key_Data payload by checking whether a payload exists during each MN power-up or application initiation.

10. The MN sends the RRQ with MIP_Key_Data MIP VSE to the PDSN.

11. The PDSN converts the RRQ to a RADIUS ARQ with MIP_Key_Data RADIUS VSA and forwards the ARQ to the home RADIUS AAA Server. The MSID is included in the ARQ.

12. The RADIUS AAA Server compares the authenticated MSID (sent from the PDSN) with the MSID in its subscriber database (associated with the NAI). If MSIDPDSN = MSIDAAA, the RADIUS AAA server, using the Public Key Identifier, determines the appropriate RSA Private key and decrypts the encrypted portion of the MIP_Key_Data payload. The RADIUS AAA Server verifies the MN-AAA Authentication Extension Authenticator using the decrypted MN_AAA key. If successful, the RADIUS AAA Server updates the subscriber profile with the decrypted MN_AAA key, MN_HA key, and CHAP key. The RADIUS AAA Server sets the AAA MIP Update State Field to KEYS UPDATED (2).

13. The RADIUS AAA Server sends an Access Reject with AAA_Authenticator RADIUS VSA to the PDSN.

14. The PDSN converts the Access Reject to a MIP RRP with AAA_Authenticator MIP VSE. RRP Code = 89 (Vendor Specific).

15. If AAA_AuthenticatorMN = AAA_AuthenticatorAAA, the MN assigns MN_AAATEMP to MN_AAA key and MN_HATEMP to MN_HA key (MN MIP Update State = KEYS VALID). Otherwise, the MN discards the temporary keys.

16. The MN initiates a new RRQ that is converted to an ARQ by the PDSN and forwarded to the RADIUS AAA Server.

17. The RADIUS AAA Server verifies the MN-AAA Authentication Extension and sets the AAA MIP Update State Field to KEYS VALID (0). The RADIUS AAA Server sends an Access Accept to the PDSN/FA.

18. The PDSN/FA sends the RRQ to the Home Agent (HA).

19. The HA sends an Access Request to the RADIUS AAA Server. The RADIUS AAA Server sends an Access Accept to the HA with the MN_HA key. The HA verifies the Mobile-Home Authentication Extension using the MN_HA key.

20. The HA sends an RRP to the PDSN/FA, which forwards the RRP to the MN. RRP Code = 0 (Success).

5. DMU Procedure Failure Operation

To improve the robustness of the DMU Procedure to account for interruptions due to UDP message loss, RRQ retransmission, or MN failure, the RADIUS AAA Server MUST maintain a MIP Update State Field, for each subscription, in one of three states (0 = KEYS VALID, 1 = UPDATE KEYS, 2 = KEYS UPDATED).

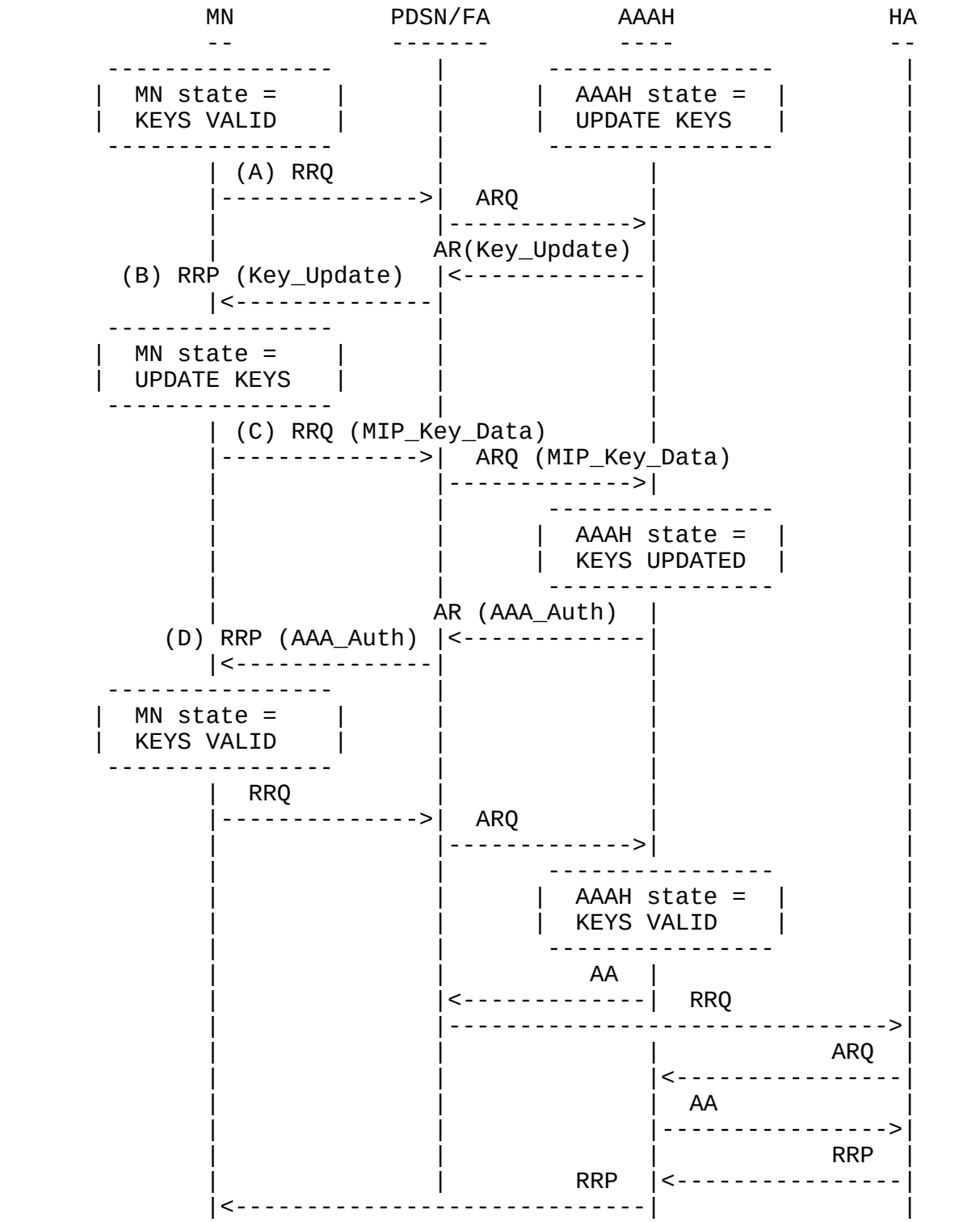

Figure 5. DMU Failure Call Flow with MN and AAA States

Each step in Figure 5 is described as follows:

1. If (A) is lost, the MN retransmits (A). The RADIUS AAA server expects (A). If the AAA server is in the UPDATE KEYS state, the RADIUS AAA Server sends AR with MIP_Key_Update_Request VSA, and the PDSN/FA sends (B).

2. If (B) is lost, the MN retransmits (A). The RADIUS AAA server expects (C). If it receives (A), the RADIUS AAA Server sends AR with MIP_Key_Update_Request VSA, and the PDSN/FA retransmits (B).

3. If (C) is lost, the mobile retransmits (C). The RADIUS AAA server expects (C) and updates the MIP keys appropriately. The RADIUS AAA server transitions to KEYS UPDATED and commits the MIP_Key_Data. The RADIUS AAA Server sends the AR with AAA_Authenticator VSA, and the PDSN/FA replies to the MN with (D).

4. If (D) is lost, the mobile retransmits (C) using the same key data sent previously. The RADIUS AAA server expects (A) using the same keys.

   a. If the RADIUS AAA server receives (C) with the same keys it received previously, it retransmits the AR with AAA_Authenticator VSA and the PDSN replies with (D), containing the AAA_Authenticator.

   b. If the RADIUS AAA server receives (C) with different keys than it received previously, the RADIUS AAA Server sends AR with MIP_Key_Update_Request VSA, the PDSN/FA retransmits (B), and the RADIUS AAA server transitions to UPDATE KEYS.

   c. If the RADIUS AAA server receives (A), which fails authentication using the keys sent in (C), the RADIUS AAA Server sends AR with MIP_Key_Update_Request, the PDSN/FA retransmits (B), and the RADIUS AAA server transitions to UPDATE KEYS.

5. Once the PDSN/FA receives (A), forwards the ARQ to the RADIUS AAA server, and the MN-AAA Authenticator is verified using the MN_AAA key, the RADIUS AAA Server transitions to the KEYS VALID state and the DMU process is complete.

The AAA DMU state machine is described in Figure 6.

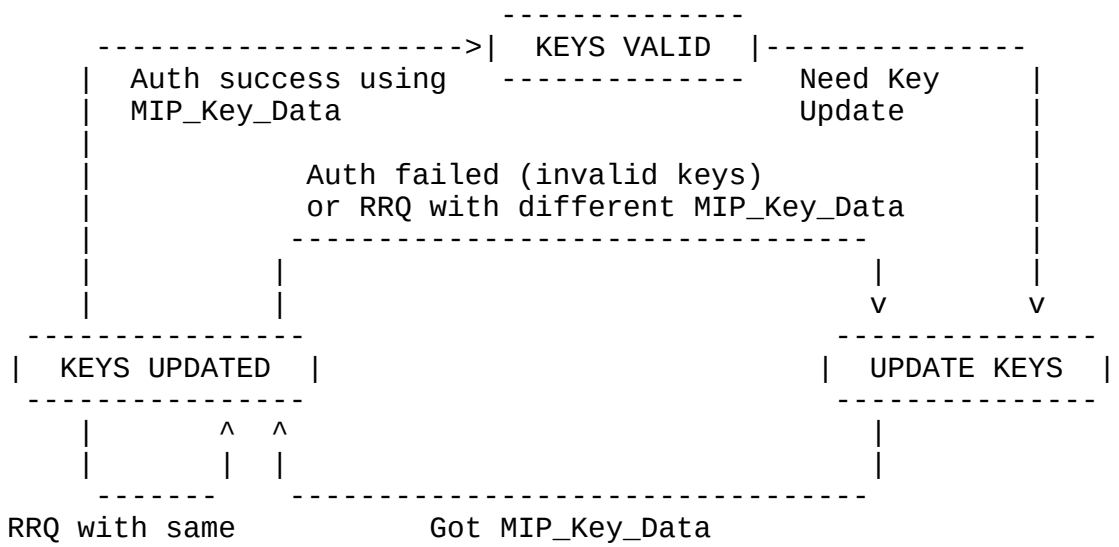

Figure 6. RADIUS AAA Server DMU State Machine

6. cdma2000(R) HRPD/1xEV-DO Support

Because the DMU Procedure occurs at the IP Layer, the DMU Procedure supports MIP key distribution in either the cdma2000(R) 1X or HRPD/1xEV-DO network. Because the cdma2000(R) HRPD/1xEV-DO network does not provide Radio Access Network (RAN) authentication, the DMU Procedure is more susceptible to a false MN attack (than in an cdma2000(R) 1X network with Cellular Authentication and Voice Encryption (CAVE) RAN authentication). For this reason, the DMU Procedure has the capability to optionally support device-to-network authentication using the MN_Authenticator.

The method of MN_Authenticator delivery to the RADIUS AAA server is outside the scope of this document, allowing service providers the flexibility to determine the most efficient/least intrusive procedure to support MN authentication during the DMU Procedure.

6.1. RADIUS AAA Support

The RADIUS AAA server MUST support three MN_Authenticator options:

1. Ignore MN_Authenticator

   Depending on other potential authentication/fraud prevention options (outside the scope of the DMU Procedure), the RADIUS AAA

Server MUST have the capability to ignore the MN_Authenticator. For example, when the RADIUS AAA Server decrypts the MIP_Key_Data payload, the AAA Server silently discards the MN_Authenticator.

2. Pre-Update Validation

   Prior to updating a subscription profile with the delivered MIP keys, the RADIUS AAA Server MUST compare the MN_AuthenticatorMN (delivered via the encrypted MIP_Key_Data payload) with the MN_AuthenticatorAAA (possibly delivered via the service provider customer care or billing/provisioning system).

3. Post-Update Validation

   After the DMU Procedure is complete, the RADIUS AAA Server stores the delivered MN_AuthenticatorMN and waits for delivery of the MN_AuthenticatorAAA (via Customer Care, interactive voice response (IVR), or some other unspecified process). Once the MN_Authenticator is delivered to the RADIUS AAA Server, the AAA MUST compare the MN_AuthenticatorMN (delivered via the encrypted MIP_Key_Data payload) with the MN_AuthenticatorAAA. If the Authenticators match, the RADIUS AAA Server authorizes access and final update of the MIP keys.

6.2. MN Support

The Mobile Node (MN) MUST store the 24-bit MN_Authenticator.

The MN MUST display the MN_Authenticator as an 8 decimal digit number (via LCD display on a handset or via a GUI for a modem). If the MN resides within a handset, the user MAY display the MN_Authenticator using the following keypad sequence: "FCN + * + * + M + I + P + RCL". Otherwise, the MN MUST display the MN_Authenticator via the device's GUI.

The MN MUST have the capability to reset the MN_Authenticator. In other words, the MN MUST have the capability to randomly/pseudo-randomly generate a new 24-bit MN_Authenticator upon user command, in accordance with RFC 4086. The reset feature mitigates possible compromise of the MN_Authenticator during shipment/storage. If the MN resides within a handset, the user MAY reset the MN_Authenticator using the following keypad sequence: "FCN + * + * + M + I + P + C + C + RCL". Otherwise, the MN MUST reset the MN_Authenticator via the device's GUI.

The MN manufacturer MAY pre-load the MN with the MN_Authenticator. For example, by pre-loading the MN_Authenticator and affixing a sticker with the MN_Authenticator (8 decimal digit representation) to

the MN (e.g., modem), the point-of-sale representative does not have to retrieve the MN_Authenticator from the MN interface.

[Optional] The MN MAY maintain a separate primary and secondary queue of MN_Authenticator/MIP_Key_Data Payload pairs. When the MN user resets the primary MN_Authenticator, the MN discards the primary MN_Authenticator (and any associated MIP_Key_Data Payload) and assigns the MN_Authenticator in the secondary queue as the primary MN_Authenticator (and assigns any associated MIP_Key_Data Payloads to the primary queue). This feature enables the user/provisioner to reset the MN_Authenticator and immediately initiate the DMU procedure without losing the MIP_Key_Data Payload pre-encryption advantage. Upon MN_Authenticator transfer from the secondary to primary queue, the MN MUST generate a new MN_Authenticator and associated MIP_Key_Data Payload for the secondary queue. The MN MUST check both the primary and secondary MN_Authenticator/MIP_Key_Data Payload queues upon power-up or application initiation. The MN MUST maintain at least one MN_Authenticator/MIP_Key_Data Payload pair in each queue.

6.3. Informative: MN_Authenticator Support

MN authentication using the MN_Authenticator gives the service provider the maximum flexibility in determining how to deliver the MN_Authenticator to the RADIUS AAA Server. The method of MN_Authenticator delivery is outside the scope of this document.

However, to provide some context as to how the MN_Authenticator may support MN authentication/fraud prevention in the HRPD/1xEV-DO environment, we describe the following possible provisioning scenario.

When a subscriber initially acquires their HRPD/1xEV-DO device and service, the point-of-sale representative records the subscription information into the billing/provision system via a computer terminal at the point-of-sale. The billing/provisioning system delivers certain information to the RADIUS AAA Server (e.g., NAI, MSID, Electronic Serial Number (ESN)) including the MN_Authenticator, which the point-of-sale representative retrieves via the MN device’s display. In the case of a modem, the manufacturer may have pre-loaded the MN_Authenticator and placed a copy of the MN_Authenticator on a sticker attached to the modem. The point-of-sale representative simply copies the 8 decimal digit value of the MN_Authenticator into the customer profile. Once the MN is loaded with the proper NAI and powered-up, the MN initiates the DMU Procedure with the RADIUS AAA Server. The RADIUS AAA Server compares the MN-delivered MN_Authenticator with the billing-system-delivered MN_Authenticator. If the authenticators match, the RADIUS AAA Server updates the

subscriber profile with the delivered MIP keys and authorizes service. If the Post-Update option is enabled within the RADIUS AAA Server, the RADIUS AAA Server tentatively updates the subscription profile until it receives the MN_Authenticator via the billing/provision system.

As another option, the service provider MAY use an IVR system in which the HRPD/1xEV-DO subscriber calls a provisioning number and inputs the MN_Authenticator. The IVR system then delivers the MN_Authenticator to the RADIUS AAA Server for final validation and Packet Data Access.

7. Security Considerations

The DMU Procedure is designed to maximize the efficiency of MIP key distribution while providing adequate key distribution security. The following provides a description of potential security vulnerabilities and their relative risk to the DMU Procedure:

7.1. Cryptographic Key Generation by the MN

Because the MN is required to properly generate the MN_AAA, MN_HA, and CHAP key, the MN must perform cryptographic key generation in accordance with accepted random/pseudo-random number generation procedures. MN manufacturers MUST comply with RFC 4086 [12] guidelines, and service providers SHOULD ensure that manufacturers implement acceptable key generation procedures. The use of predictable cryptographic keys could be devastating to MIP security. However, the risk of not using acceptable random/pseudo-random key generation is minimal as long as MN manufacturers adhere to RFC 4086 guidelines. Furthermore, if a key generation flaw is identified, the flaw appears readily correctable via a software patch, minimizing the impact.

7.2. Man-in-the-Middle Attack

The DMU procedure is susceptible to a Man-in-the-Middle (MITM) attack; however, such an attack appears relatively complex and expensive. When Authentication and Key Agreement (AKA) is deployed within cdma2000(R) 1X, the MITM Attack will be eliminated. The risk of an MITM Attack is minimal due to required expertise, attack expense, and impending cdma2000(R) 1X mutual authentication protection. If a particular cdma2000(R) 1X network does not support A-key authentication, the MN_Authenticator MAY optionally be used.

7.3. RSA Private Key Compromise

Because one RSA Private key may be associated with millions of MNs (RSA public key), it is important to protect the RSA Private key from disclosure to unauthorized parties. If a MN manufacturer is generating the RSA public/private key pair, the MN manufacturer MUST establish adequate security procedures/policies regarding the dissemination of the RSA Private key to the appropriate service provider. An RSA Private key SHOULD be distributed to a legitimate cdma2000(R) service provider only. If a service provider is generating their own RSA public/private key pair, the service provider MUST protect the RSA Private key from disclosure to unauthorized parties.

7.4. RSA Encryption

Several vulnerabilities have been identified in certain implementations of RSA; however, they do not appear applicable to the DMU Procedure.

7.5. False Base Station/PDSN

The MN appears to be protected against a false BS denial-of-service (DOS) attack, since only the proper RADIUS AAA server can recover the AAA_Authenticator. This method of preventing a false base station attack assumes security of the network messaging between the AAA and the serving system, as discussed in Section 7.9.

7.6. cdma2000(R) 1X False MN

The cdma2000(R) 1X network appears adequately protected against a false MN by IS-2000 challenge-response authentication. If DMU is used outside the cellular domain, equivalent authentication procedures are required for the same level of security.

7.7. HRPD/1xEV-DO False MN

The 1xEV-DO RADIUS AAA Server MAY optionally authenticate the MN using the MN_Authenticator to prevent a fraudulent MN activation.

7.8. Key Lifetimes

There is no explicit lifetime for the keys distributed by DMU.

The lifetime of the keys distributed by DMU is determined by the system operator through the RADIUS AAA server. The MN_AAA and MN_HA key lifetimes can be controlled by initiating an update as needed.

Furthermore, the DMU process is protected against false initiation because the MN cannot initiate DMU. This makes it unworkable to provide an explicit lifetime to the MN, since the MN cannot take any action to renew the keys after expiration.

7.9. Network Message Security

The security of the MN-HA keys delivered from the RADIUS AAA server to the MIP home agent requires confidentiality for network messages containing such keys. The specification of security requirements for network messages is the responsibility of the operator, and is outside the scope of this document. (Note that similar considerations apply to the distribution of Shared Secret Data, which is already transmitted between nodes in the ANSI-41 network.)

If DMU is used outside the domain of a cellular operator, RADIUS security features MAY be used, including the Request-Authenticator and Response-Authenticator fields defined in [4] and the Message-Authenticator attribute defined in [13].

8. Verizon Wireless RADIUS Attributes

Three new RADIUS Attributes are required to support the DMU Procedure and are specified as follows:

Type: 26
Length: >9
Verizon Wireless Enterprise/Vendor ID: 12951

MIP_Key_Update_Request:

The Home RADIUS AAA Server includes this attribute to indicate that MIP key update is required.

Vendor-Type = 1
Vendor-Length = 3 bytes
Vendor-Value = PKOID of the RADIUS AAA Server

MIP_Key_Data:

Key data payload containing the encrypted MN_AAA key, MN_HA key, CHAP key, MN_Authenticator, and AAA_Authenticator. This payload also contains the Public Key Identifier.

Vendor-Type = 2
Vendor-Length = 134 bytes

NOTE: Vendor-Length depends on the size of the RSA modulus. For example, when RSA-512 is used, Vendor-Length = 70 bytes.
Vendor-Value = 128 byte RSA encryption payload (when 1024-bit RSA used), which contains encrypted MN_AAA key, MN_HA key, CHAP key, MN_Authenticator, and AAA_Authenticator.
The four (4) byte Public Key Identifier is concatenated to the encrypted payload.

AAA_Authenticator:

The 64-bit AAA_Authenticator value decrypted by the Home RADIUS AAA Server.

Vendor-Type = 3
Vendor-Length = 10 bytes
Vendor-Value = decrypted AAA_Authenticator from Home RADIUS AAA Server.

Public Key Invalid:

The home RADIUS AAA Server includes this attribute to indicate that the public key used by the MN is not valid.

Vendor-Type = 4
Vendor-Length = 2 bytes
Vendor-Value = none.

Note: An Organization may define RADIUS VSAs using its own Organization identifier.

9. Verizon Wireless Mobile IP Extensions

Three Verizon Wireless Mobile IP Vendor/Organization-Specific Extensions (VSEs) (RFC 3115), required to support the DMU Procedure, are specified as follows:

Type: 38 (CVSE-TYPE-NUMBER)

Verizon Wireless Vendor ID: 12951 (high-order octet is 0 and low order octets are the SMI Network Management Private Enterprise Code of the Vendor in the network byte order, as defined by IANA).

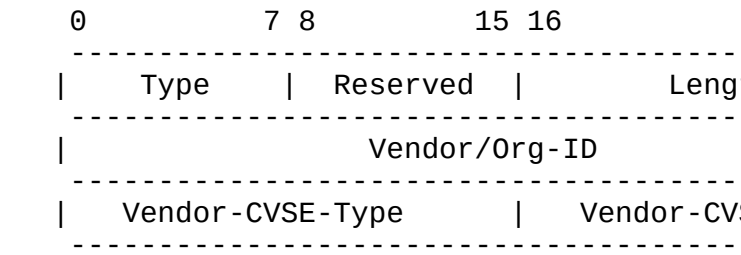

Figure 7. Critical Vendor/Organization-Specific Extension

MIP_Key_Request:

The Home RADIUS AAA Server includes this extension to indicate that MIP key update is required.

Length = 7

NOTE: The RFC 3115 Editor has stated that the Reserved field is not included in the length determination.
Vendor-CVSE-Type = 1
Vendor-CVSE-Value = PKOID sent in the RADIUS MIP_Key_Update_Request attribute.

MIP_Key_Data:

Key data payload containing encrypted MN_AAA key, MN_HA key, CHAP key, MN_Authenticator, and AAA_Authenticator. This payload also contains the Public Key Identifier.

Length = 138
NOTE: Length depends on the size of the RSA modulus. For example, when RSA-512 is used, Length = 74 bytes.
Vendor-CVSE-Type = 2
Vendor-CVSE-Value = 128 byte RSA encryption payload (when 1024-bit RSA used) which contains encrypted MN_AAA key, MN_HA key, CHAP key, MN_Authenticator, and AAA_Authenticator.
The four (4) byte Public Key Identifier and DMUV is concatenated to the encrypted payload.

AAA_Authenticator:

The 64-bit AAA_Authenticator value decrypted by the Home RADIUS AAA Server.

Length = 14 bytes
Vendor-CVSE-Type = 3
Vendor-CVSE-Value = decrypted AAA_Authenticator from the Home RADIUS AAA Server.

Public Key Invalid:

The Home RADIUS AAA Server includes this extension to indicate that the public key used by the MN is not valid.

Length = 6 bytes
Vendor-CVSE-Type = 4
Vendor-CVSE-Value = none.

Note: An Organization may define VSEs using their own Organization identifier.

10. Public Key Identifier and DMU Version

The Public Key Identifier (Pub_Key_ID) is used during the Dynamic Mobile IP Update (DMU) procedure to allow the RADIUS AAA Server to distinguish between different public keys (which may be assigned by different manufacturers, service providers, or other organizations). The Public Key Identifier consists of the PKOID, PKOI, PK_Identifier, and ATV fields. The DMU Version field enables subsequent revisions of the DMU procedure.

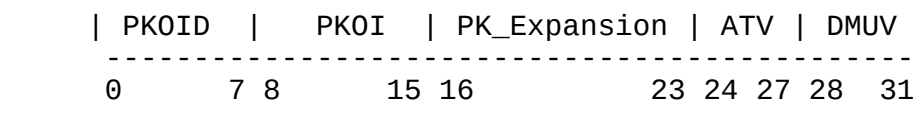

Figure 8. Public Key Identifier and DMUV

Each Public Key Organization (PKO) MUST be assigned a Public Key Organization Identifier (PKOID) to enable the RADIUS AAA Server to distinguish between different public keys created by different PKOs (see Table 1).

If a service provider does not provide the MN manufacturer with a (RSA) public key, the manufacturer MUST generate a unique RSA Public/Private key pair and pre-load each MN with the RSA public key (1024-bit modulus by default). The manufacturer MAY share the same RSA Private key with multiple service providers as long as reasonable security procedures are established and maintained (by the manufacturer) to prevent disclosure of the RSA Private (decryption) key to an unauthorized party.

The Public Key Organization Index (PKOI) is an 8-bit field whose value is defined at the discretion of the PKO. For example, a device manufacturer MAY incrementally assign a new PKOI for each Public/Private key pair when the pair is created.

The PK_Expansion field enables support for additional PKOs or expansion of the PKOI.

The DMU Version field allows for DMU Procedure version identification (see Table 2).

The Algorithm Type and Version (ATV) field allows for identification of the public key algorithm and version used (see Table 3).

Table 1. Public Key Organization Identification Table

PKOID (HEX)	Public Key Organization (PKO)	PKOID (HEX)	Public Key Organization (PKO)
00	RESERVED	40	Sanyo Fisher Company
01	RESERVED	41	Sharp Laboratories of America
02	RESERVED	42	Sierra Wireless, Inc.
03	RESERVED	43	Sony Electronics
04	RESERVED	44	Synertek, Inc.
05	RESERVED	45	Tantivy Communications, Inc.
06	RESERVED	46	Tellus Technology, Inc.
07	RESERVED	47	Wherify Wireless, Inc.
08	RESERVED	48	Airbiquity
09	RESERVED	49	ArrayComm
0A	Verizon Wireless	4A	Celletra Ltd.
0B	AAPT Ltd.	4B	CIBERNET Corporation
0C	ALLTEL Communications	4C	CommWorks Corporation, a 3Com Company
0D	Angola Telecom	4D	Compaq Computer Corporation
0E	Bell Mobility	4E	ETRI
0F	BellSouth International	4F	Glenayre Electronics Inc.
10	China Unicom	50	GTRAN, Inc.
11	KDDI Corporation	51	Logica
12	Himachal Futuristic Communications Ltd.	52	LSI Logic
13	Hutchison Telecom (HK), Ltd.	53	Metapath Software International, Inc.
14	IUSACELL	54	Metawave Communications
15	Komunikasi Selular Indonesia (Komselindo)	55	Openwave Systems Inc.
16	Korea Telecom Freetel, Inc.	56	ParkerVision, Inc.
17	Leap	57	QUALCOMM, Inc.
18	LG Telecom, Ltd.	58	QuickSilver Technologies
19	Mahanagar Telephone Nigam Limited (MTNL)	59	Research Institute of Telecommunication Transmission, MII (RITT)
1A	Nextel Communications, Inc.	5A	Schema, Ltd.
1B	Operadora UNEFON SA de CV	5B	SchlumbergerSema
1C	Pacific Bangladesh Telecom Limited	5C	ScoreBoard, Inc.
1D	Pegaso PCS, S.A. DE C.V.	5D	SignalSoft Corp.

PKOID (HEX)	Public Key Organization (PKO)	PKOID (HEX)	Public Key Organization (PKO)
1E	Pele-Phone Communications Ltd.	5E	SmartServ Online, Inc.
1F	Qwest	5F	TDK Corporation
20	Reliance Infocom Limited	60	Texas Instruments
21	Shinsegi Telecomm, Inc.	61	Wherify Wireless, Inc.
22	Shyam Telelink Limited	62	Acterna
23	SK Telecom	63	Anritsu Company
24	Sprint PCS	64	Ericsson
25	Tata Teleservices Ltd.	65	Grayson Wireless
26	Telecom Mobile Limited	66	LinkAir Communications, Inc.
27	Telstra Corporation Limited	67	Racal Instruments
28	Telus Mobility Cellular, Inc.	68	Rohde & Schwarz
29	US Cellular	69	Spirent Communications
2A	3G Cellular	6A	Willtech, Inc.
2B	Acer Communication & Multimedia Inc.	6B	Wireless Test Systems
2C	AirPrime, Inc.	6C	Airvana, Inc.
2D	Alpine Electronics, Inc.	6D	COM DEV Wireless
2E	Audiovox Communications Corporation	6E	Conductus, Inc.
2F	DENSO Wireless	6F	Glenayre Electronics Inc.
30	Ditrans Corporation	70	Hitachi Telecom (USA), Inc.
31	Fujitsu Network Communication, Inc.	71	Hyundai Syscomm Inc.
32	Gemplus Corporation	72	ISCO
33	Giga Telecom Inc.	73	LG Electronics, Inc.
34	Hyundai CURITEL, Inc.	74	LinkAir Communications, Inc.
35	InnovICs Corp	75	Lucent Technologies, Inc.
36	Kyocera Corporation	76	Motorola CIG
37	LG Electronics, Inc.	77	Nortel Networks
38	LinkAir Communications, Inc.	78	Repeater Technologies
39	Motorola, Inc.	79	Samsung Electronics Co., Ltd.
3A	Nokia Corporation	7A	Starent Networks
3B	Novatel Wireless, Inc.	7B	Tahoe Networks, Inc.
3C	OKI Network Technologies	7C	Tantivy Communications, Inc.

PKOID (HEX)	Public Key Organization (PKO)	PKOID (HEX)	Public Key Organization (PKO)
3D	Pixo	7D	WaterCove Networks
3E	Research In Motion	7E	Winphoria Networks, Inc.
3F	Samsung Electronics Co., Ltd.	7F	ZTE Corporation

Note: 80 through FF will be assigned by the PKOID administrator (Verizion Wireless).

Table 2. DMU Version

DMU Version Value	DMU Version
00	RFC 4784
01	Reserved
02	Reserved
03	Reserved
04	Reserved
05	Reserved
06	Reserved
07	Cleartext Mode

Table 3. Algorithm Type and Version

ATV Value	Public Key Algorithm Type and Version
00	Reserved
01	RSA - 1024
02	RSA - 768
03	RSA - 2048
04	Reserved
05	Reserved
06	Reserved
07	Reserved

11. Conclusion

The Dynamic Mobile IP Key Update (DMU) Procedure enables the efficient, yet secure, delivery of critical Mobile IP cryptographic keys. The use of cryptographic keys (and hence, the bootstrapping of such MIP keys using the DMU Procedure) is essential to commercial delivery of Mobile IP service in cdma2000 1xRTT and HRPD/1xEV-DO networks or other networks that utilize Mobile IP.

12. Normative References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

13. Informative References

[2] TIA/EIA/IS-2000 Series, Revision A, Telecommunications Industry Association, March 2000.

[3] TIA/EIA/IS-856, cdma2000(R) High Rate Packet Data Air Interface Specification, Telecommunications Industry Association, November 2000.

[4] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.

[5] Calhoun, P., Loughney, J., Guttman, E., Zorn, G., and J. Arkko, "Diameter Base Protocol", RFC 3588, September 2003.

[6] TIA/EIA/IS-835-A, cdma2000(R) Wireless IP Network Standard, Telecommunications Industry Association, May 2001.

[7] ANSI/TIA/EIA-41-D-97, Cellular Radiotelecommunications Intersystem Operations, Telecommunications Industry Association, December 1997

[8] ANSI/TIA/EIA-683-B-2001, Over-the-Air Service Provisioning of Mobile Stations in Spread Spectrum Systems, Telecommunications Industry Association, December 2001

[9] Jonsson, J. and B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", RFC 3447, February 2003.

[10] Dommety, G. and K. Leung, "Mobile IP Vendor/Organization-Specific Extensions", RFC 3115, April 2001.

[11] TIA-2001-A, Interoperability Specifications (IOS) for cdma2000(R) Access Network Interfaces, Telecommunications Industry Association, August 2001.

[12] Eastlake, D., 3rd, Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

[13] Rigney, C., Willats, W., and P. Calhoun, "RADIUS Extensions", RFC 2869, June 2000.

14. Acknowledgments

Thanks to Jeffrey Dyck (Qualcomm), James Willkie (Qualcomm), Jayanth Mandayam (Qualcomm), Marcello Lioy (Qualcomm), Michael Borella (CommWorks), Cliff Randall (CommWorks), Daniel Cassinelli (CommWorks), Edward Dunn (CommWorks), Suresh Sarvepalli (CommWorks), Gabriella Ambramovici (Lucent), Semyon Mizikovsky (Lucent), Sarvar Patel (Lucent), Peter McCann (Lucent), Ganapathy Sundaram (Lucent), Girish Patel (Nortel), Glen Baxley (Nortel), Diane Thompson (Ericsson), Brian Hickman (Ericsson), Somsay Sychaleun (Bridgewater), Parm Sandhu (Sierra Wireless), Iulian Mucano (Sierra Wireless), and Samy Touati (Ericsson) for their useful discussions and comments.

Appendix A: Cleartext-Mode Operation

DMU supports a cleartext mode for development testing where DMUV = 7. The MIP_Key_Data payload will assume the same size as if RSA 1024-bit encryption were applied to the payload. In this mode, the MIP_Key_Data RADIUS Attribute and MIP Vendor Specific Extension will be 134 bytes and 138 bytes in length, respectively. Thus, in cleartext mode, the payload MUST consist of 48 bytes of keys (MN_AAA, MN_HA, and CHAP key), 8-byte AAA_Authenticator, 3-byte MN_Authenticator. The next 69 bytes will be padded with "0" bits.

MIP_Key_Data = MN_AAAH key, MN_HA key, CHAP_key, MN_Authenticator, AAA_Authenticator, Padding (69 bytes), Public_Key_IDi, DMUV

Where:

MN_AAA key = 128-bit random MN / RADIUS AAA Server key.

MN_HA key = 128-bit random MN / Home Agent (HA) key.

CHAP_key = 128-bit random Simple IP authentication key.

MN_Authenticator = 24-bit random number.

AAA_Authenticator = 64-bit random number used by MN to authenticate the RADIUS AAA Server.

Padding = 69 bytes of 0’s.

DMU Version (DMUV) = 4-bit identifier of DMU version.

Public Key Identifier (Pub _Key_ID) = PKOID, PKOI, PK_Expansion, ATV

Where:

Public Key Organization Identifier (PKOID) = 8-bit serial number identifier of the Public Key Organization (PKO) that created the Public Key.

Public Key Organization Index (PKOI) = 8-bit serial number used at PKO discretion to distinguish different Public/Private key pairs.

PK_Expansion = 8-bit field to enable possible expansion of PKOID or PKOI fields. (Note: Default value = 0xFF)

Algorithm Type and Version (ATV) = 4-bit identifier of the algorithm used.

Authors' Addresses

Christopher Carroll*
Ropes & Gray LLP
Fish & Neave IP Group
One International Place
Boston, MA 02110

Phone: 617-951-7756
EMail: Christopher.Carroll@ropesgray.com

* This document was developed while at Verizon Wireless.

Frank Quick
Qualcomm Incorporated
5775 Morehouse Drive
San Diego, CA 92121 USA

Phone: 858-658-3608
EMail: fquick@qualcomm.com

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.