Core
Internet-Draft
Intended status: Standards Track
Expires: October 26, 2012

B. Sarikaya
Huawei USA
Y. Ohba
Toshiba
R. Moskowitz
Verizon Business Systems
Z. Cao
China Mobile
R. Cragie
Pacific Gas and Electric
April 24, 2012

Security Bootstrapping Solution for Resource-Constrained Devices
draft-sarikaya-core-sbootstrapping-04

Abstract

This document describes how to initially configure the network of resource constrained nodes securely, a.k.a., security bootstrapping. Bootstrapping architecture, communication channel and bootstrap security methods are described. System level objectives for security bootstrapping are stated followed by the protocols that can be used. Bootstrapping solution is based on EAP-TLS authentication with the use of raw public keys used as certificates.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 26, 2012.

Table of Contents

1. Introduction

Bootstrapping is any processing required before the network can operate. Typically this will require a number of settings to be transferred between nodes at all layers. This could include anything from link-layer information (i.e., wireless channels, link-layer encryption keys) to application-layer information (i.e., network names, application encryption keys).

Bootstrapping is complete when settings have been securely transferred prior to normal operation in the network.

The bootstrapping problem is not specific to any MAC or PHY. This problem exists across any two nodes which have no previous knowledge of each other. In particular, this problem is complicated when the nodes are resource-constrained and may not have an advanced user interface. The IETF is instrumental in defining standards which will be used by The Internet of Things (IOT). Ensuring these standards can be used across nodes and networks requires some form of bootstrapping which any node can use.

Existing standards will be used as much as possible in this document. The method proposed here should work across many different underlying layers. It could be used to allow two nodes on the same physical network to join at the physical layer, or allow two nodes on an incompatible physical network to join at the IPv6 layer.

The document continues in Section 2 on bootstrapping architecture, in Section 3 on allowable security methods in bootstrapping, in Section 4 on system level objectives of secure bootstrapping, and Section 5 on secure bootstrapping solution. Appendix A is on examples of node configuration, Appendix B is on samples of exchanges during bootstrapping and Appendix C is on the protocols used to carry EAP in link layer/IP layer including HIP-DEX.

2. Bootstrapping Architecture

2.1. Areas of Boostrapping

In order to provide a flexible architecture, the bootstrapping method is split into five distinct areas and two distinct phases. The five areas are a 'user interface', 'bootstrap profile', 'security method', 'bootstrap protocol', and the 'communications channel'.

The phases are provisioning phase and bootstrapping phase. In the provisioning phase, statically configured parameters (e.g., certificates) needed for the bootstrapping phase is provisioned. In

the bootstrapping phase, dynamically configured information is set up using the statically configured information provided in the provisioning phase.

The user interface provides both user input and user output. Simple nodes may only have a push-button and LED, more complex nodes may have a graphical display and keyboard. The user interface (which could be implemented as network management software graphical user interface running at the remote end) provides interaction between the user and bootstrapping methods. The user interface would be used during bootstrapping as an OOB channel. It may also be used to specify bootstrapping policies.

The user interface provides the interaction between the user and the bootstrap protocol. The user interface will vary depending on the capabilities of the node. Examples might include a push-button and LED on simple nodes, to full-blown graphical user interfaces. Note that a 'bootstrapping tool' used to initially deploy a network is just a special user interface. This allows a very uniform protocol in deployment and use of networks.

User interface is out-of-scope and will not be further discussed.

Two nodes communicate through some channel. For our purposes this is split into the 'control channel' and 'data channel'. The control channel is used for the bootstrap protocol, and the data channel is used during normal network operation. A node may support multiple control or data channels. When the control and data channels are the same, the bootstrapping is done In Band (IB). When the control and data channels are different, the bootstrapping is performed Out Of Band (OOB). An 802.15.4 network for instance would use an 802.15.4 control channel for IB bootstrapping, but a control channel of perhaps IrDA or USB for OOB bootstrapping.

The 'bootstrap profile', i.e. statically configured parameters during the provisioning phase, defines what information should be exchanged during the process. A single node may run the protocol multiple times with different profiles. If the user wishes to associate a new lightswitch, the protocol is first run with the '802.15.4 Wireless Profile', through which it learns the channel and PAN-ID. The node then runs a 'Security Exchange Profile' to learn the needed encryption keys. Finally it runs a 'Lightswitch Association Profile' through which it learns which light to associate with.

An example of the 'bootstrap profile' attribute is the 'administrative domain name'. 'Bootstrap profiles' are required to be modified when the corresponding administrative domains are changed, a.k.a. recommissioning. In recommissioning, the domains are

administratively repartitioned and nodes are therefore recommissioned.

The 'security method' defines supported security methods for bootstrapping. The supported security methods will depend on the control channel and bootstrap profile. In one node if the control channel is secure, then a simple clear-text security method is supported. For example when a physical connection between two nodes is used, the control channel is considered secure. However when the control channel is not secure, this clear-text security method is not supported. The 'bootstrap profile' additionally defines allowed security methods. Higher security nodes may outlaw ever performing a clear-text exchange, even if the control channel is deemed secure.

The 'bootstrap protocol' defines the actual messages exchanged during bootstrapping. The messages are used to transfer between nodes data, node information, and network state. The selected security method runs on top of the control channel, such as EAP-GPSK etc.

2.2. Architecture

Security bootstrapping architecture is structured in a hierarchy of nodes going from the least resource constraint to the most resource constraint. At the top there is a root node. The root node is called Coordinator or Trust Center in Zigbee and 6LowPAN Border Router (6LBR) in 6LoWPAN ND.

At the next level there are interior Routers. Routers are able to run a routing protocol between other routers and the root. Routers are called 6LowPAN Routers (6BR) in 6LoWPAN ND.

At the lowest level there are the nodes. The nodes do not run a routing protocol. They can connect to the nearest router over a single radio link. The nodes are called End Devices in Zigbee and hosts in 6LoWPAN ND.

Routers first join the network as a node and go through security bootstrapping operations in order to create a Master Session Key (MSK). Next, routers execute routing protocol, e.g. [I-D.ietf-roll-rpl] specific steps to create session keys with their neighbors and to establish upstream and downstream next hop parents.

At each node hierarachy level described above, there are lower-layer and higher-layer protocols to bootstrap their ciphering keys, where the lower-layer refers to layers below IP layer including IEEE 802.15.4 MAC layer and LoWPAN adaptation layer and the higher-layer refers to IP layer and the above. In general, required bootstrapping procedures depend on the bootstrapping protocols to use. Section

Section 5 describes the bootstrapping procedures where EAP (Extensible Authentication Protocol) [RFC3748] and other protocols are used as the bootstrapping protocols.

3. Bootstrap Security Method

The bootstrap security method defines allowable security methods. A node may choose to support or use a subset of these methods. This is NOT the security architecture used for the application, but only the security used during bootstrapping. Typically some high-security method is used to generate a shared secret, which then switches to simplier symmetric encryption to secure the actual bootstrapping channel. The techniques negotiated should take advantage of hardware resources available, such as hardware encryption accelerators on an end node.

3.1. None

This is the simplist security method. No encryption or authentication is provided, messages are exchanged completely in clear-text. It is assumed some other layer provides security, such as a physical connection between devices.

3.2. Asymmetric with User Authentication, Followed by Symmetric

A Diffie-Hellman style key exchange is used to generate a shared secret. The authentication will be provided by the user, by confirming cryptographic signatures between two devices. With the shared secret generated through the DH, some symmetric encryption is used to secure the actual bootstrapping channel.

3.3. Asymmetric with Certificate Authority, Followed by Symmetric

Public key exchanges are used (aka: DH again), but with a Certificate Authority. Once a shared secret exists, symmetric encryption is used to secure the actual bootstrapping channel.

3.4. Cryptographically Generated Address Based Address Ownership Verification

A node may generate the global unique address using different techniques other than the stateless address autoconfiguration. For example, Cryptographically Generated Addresses (CGA) [RFC3972] is a type of global unique address that can be used to verify the address ownership. When the node uses CGA, it MUST execute SeND protocol [RFC3971]. In a 6LOWPAN network, a modified 6LOWPAN ND Protocol [I-D.ietf-6lowpan-nd] must be executed between the node and the edge

router.

4. Secure Bootstrapping System Level Objectives

Authentication/ reauthentication: nodes joining the network MUST at the first place authenticate to the trust center. In order to achieve secure multi-hop routing, the node MUST authenticate to its upstream and downstream neighbors. A bootstrapping solution MUST support re-authentication of resource-constrained devices and re-keying of dynamically generated keys.

Data Confidentiality: the data communication between two endpoints MAY be encrypted using the derived key, avoiding being eavesdropped by a non-trusted third part.

Data Integrity: the data communication between two endpoints MUST NOT be altered by some intermediate nodes. The nodes should be able to detect the non-integral data.

Keys and key freshness: the keys used for data communication MUST have a lifetime, in order to keep their freshness. A bootstrapping solution MUST support both symmetric and asymmetric key authentication. If distribution of a key to be used for a resource-constrained device is required, a bootstrapping solution MUST support secure key distribution to prevent the key from eavesdropping, alternation and replay attacks.

Multi domain support: A bootstrapping solution MUST be able to allow resource-constrained devices that may be subscribed to different administrative domains to be connected to the same access network at the same time.

Multi domain support: A bootstrapping solution MUST be able to allow resource-constrained devices to be recommissioned. Recommissioning a device is defined to be (1) an resource-constrained device is administratively switched to a different domain, or (2) acting a new role with a different function set, or (3) both administrative domain and function set are modified.

Identities: A bootstrapping solution MUST be able to allow a resource-constrained device to use various types of identities used for authentication, including device identities, user identities or combinations of different types of identities. Also a bootstrapping solution MUST be able to allow a resource-constrained device to change its identities used for authentication over time.

Authentication infrastructure: A bootstrapping solution MUST be able

to operate with or without an authentication infrastructure.

5. Secure Bootstrapping Solution using Raw Public Keys

When a new resource-constrained device is deployed, it configures its global unique IPv6 address first. This is done by 6LoWPAN Neighbor Discovery (6LoWPAN-ND)'s Router Solicitation/Router Advertisement message exchange [I-D.ietf-6lowpan-nd]. The newly generated IPv6 address can not be used until the joining device is authenticated and securely joins the network. After the authentication, the joining device receives the current group key of the network, so that the IPv6 registration and further communication can be protected by the link layer ciphering e.g. 802.15.4, then it can start using its global unique IPv6 address for communication.

For authentication, Extensible Authentication Protocol (EAP) MUST be used. EAP authentication framework is explained in Appendix C.1.

The EAP method EAP-TLS [RFC5216] can be used for the resource-constrained device authentication. Instead of X.509 certificates, raw public key of the device MUST be used. EAP-TLS is executed between the joining device and the AAA server which acts as the Authentication Server (AS). After a successful authentication, the device and the AAA server establish a Master Session Key (MSK), and then the AAA server exports the MSK to the authenticator. Upon receipt of the MSK, the authenticator distributes the group key to the joining device within the authentication success message. The group key is encrypted by a Key Encryption Key derived from the MSK.

The resource-constrained device initiates the EAP authentication process by sending a message of initiation to the authenticator, i.e. the root node or 6LBR. The root node requests the identity from the device. The identity information includes the device's network access ID (NAI). When the root node receives NAI of the device, it sends the identity information to the AS.

The AS starts the EAP-TLS authentication process by sending a EAP-Request/TLS-Start to the device. The device generates a client random number and responds with an EAP-Response/TLS-Client-Hello message which contains the TLS version, a client random number, a set of cipher suites. Only one cipher suite MUST be offered in Client-Hello message with RC4-SHA1.

The device MUST add an extension of type cert_type defined in [I-D.ietf-tls-oob-pubkey] to Client-Hello message. This type MUST be set to RawPublicKey.

Upon receipt of Client Hello, if the AS supports raw public key extension, it generates a server random number, a new session ID, and includes only the SubjectPublicKeyInfo part of the certificate, rather than the whole certificate in the Certificate message and then sends them to the device with an EAP-Request/TLS-Server-Hello message.

With the client and server random number, the device generates a pre_master_secret, then sends it in Client-Key-Exchange field of EAP-Response/TLS-Client-Finished message to the AS.

The AS derives the Master Session Key (MSK) and replies with EAP-Request/TLS-Server-Finished message. The SE device also derives the MSK after receiving the Server Finished and acknowledges with EAP-Response/EAP-TLS message.

The AS then exports the MSK to the authenticator in RADIUS Access-Accept message, the authenticator subsequently sends the EAP-Success message to the SE device. The AS MUST send the group key in this message and the EAP-TLS ends.

When a device is not a direct neighbor of the authenticator, its parent node MUST act as relay. Different EAP encapsulation protocols have different mechanisms for the relay function, for details, see Appendix C.

6. Security Considerations

When security bootstrapping resource constraint nodes is undertaken, several attacks are possible and security bootstrapping methods described in this document do not protect the nodes against such attacks. These attacks are similar to the ones described in [RFC3971] and mainly stem from unsecured link layer. Link layer must be secured on each node before the node can begin security bootstrapping.

If a bootstrapping protocol does not rely on a pre-shared key for peer authentication, it must rely on an online or offline third-party (e.g., an authentication server, a key distribution center in Kerberos, a certification authority in PKI, a private key generator in ID-based cryptography and so on) to prevent man-in-the-middle attacks during peer authentication. Depending on use cases, a resource-constrained device may not always have access to an online third-party for peer authentication.

Depending on use cases, a bootstrapping protocol may deal with authorization separately from authentication in terms of timing and

signaling path. For example, two resource-constrained devices A and B may perform mutual authentication using authentication credentials provided by an offline third-party X whereas resource-constrained device A obtains authorization for running a particular application with resource-constrained device B from an online third-party Y before or after the authentication. In some use cases, authentication and authorization are tightly coupled, e.g., successful authentication also means successful authorization. A bootstrapping protocol supports various types of authentication and authorization or different bootstrapping protocols may be used for different types of authentication and authorization.

If authorization information includes cryptographic keys, a special care must be taken for dealing with the keys, e.g., guidelines for AAA-based key management are described in [RFC4962]. A recommissioning use case may require revocation and re-installation of authentication credentials (i.e., a certificate or a shared secret and identity information, etc.) to a large number of resource-constrained devices that are already deployed. Re-installation of authentication credentials must be as secure as the initial installation regardless of whether the re-installation is done manually or automatically.

If resource-constrained devices use a multicast group key for peer authentication or message authentication or encryption, the group key must be securely distributed to the current members of the group for both initial key distribution and key update. Protocols designed for group key management such as GSAKMP [RFC4535], GDOI [RFC3547] and MIKEY [RFC3830] may be used for group key distribution. Alternatively, key wrap attributes for securely encapsulating group key may be defined in network access authentication protocols such as PANA [RFC5191] and EAP-TTLSv0 [RFC5281]. Those protocols use an end-to-end, point-to-point communication channel with a pair-wise security association between a key distribution center and each key recipient. Further considerations may be needed for more efficient group key management to support a large number of resource-constrained devices.

7. IANA Considerations

This memo includes no request to IANA.

8. Contributors

The people listed below have made significant text contributions to this document.

Colin O'Flynn

colin.oflynn@atmel.com

9. Acknowledgements

Special thanks also to Rene Struik, Carsten Borman, Gary Yang, Alper Yegin and Tero Kivinen for their comments that helped us improve the writing. Discussions with Hannes Tschofenig have been very inspiring.

10. References

10.1. Normative References

[802.15.4] IEEE Std 802.15.4-2006, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs)", September 2006.

[802.1x] IEEE Std 802.1X-2010, "IEEE 802.1X Port-Based Network Access Control", February 2010.

[I-D.ietf-tls-oob-pubkey] Wouters, P., Gilmore, J., Weiler, S., Kivinen, T., and H. Tschofenig, "TLS Out-of-Band Public Key Validation", draft-ietf-tls-oob-pubkey-02 (work in progress), March 2012.

[RF4CE] ZigBee Alliance, "Zigbee RF4CE Specification Version 1.00", March 2009.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, "Extensible Authentication Protocol (EAP)", RFC 3748, June 2004.

[RFC4919] Kushalnagar, N., Montenegro, G., and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals", RFC 4919, August 2007.

[RFC5191] Forsberg, D., Ohba, Y., Patil, B., Tschofenig, H., and A. Yegin, "Protocol for Carrying Authentication for Network

Access (PANA)", RFC 5191, May 2008.

[RFC5216] Simon, D., Aboba, B., and R. Hurst, "The EAP-TLS Authentication Protocol", RFC 5216, March 2008.

[RFC5548] Dohler, M., Watteyne, T., Winter, T., and D. Barthel, "Routing Requirements for Urban Low-Power and Lossy Networks", RFC 5548, May 2009.

[RFC5673] Pister, K., Thubert, P., Dwars, S., and T. Phinney, "Industrial Routing Requirements in Low-Power and Lossy Networks", RFC 5673, October 2009.

[ROMER04] Romer, K. and F. Mattern, "The design space of wireless sensor networks", IEEE Wireless Communications, vol. 11, no. 6, pp. 54-61, December 2004.

[SE2.0] ZigBee Alliance, "Smart Energy Profile 2.0 Technical Requirements Document", April 2010.

10.2. Informative References

[C1222] American National Standard, "Protocol Specification For Interfacing to Data Communication Networks", ANSI C12.22-2008, 2008.

[I-D.ietf-6lowpan-nd]
Shelby, Z., Chakrabarti, S., and E. Nordmark, "Neighbor Discovery Optimization for Low Power and Lossy Networks (6LoWPAN)", draft-ietf-6lowpan-nd-18 (work in progress), October 2011.

[I-D.ietf-core-coap]
Shelby, Z., Hartke, K., Bormann, C., and B. Frank, "Constrained Application Protocol (CoAP)", draft-ietf-core-coap-09 (work in progress), March 2012.

[I-D.ietf-roll-rpl]
Brandt, A., Vasseur, J., Hui, J., Pister, K., Thubert, P., Levis, P., Struik, R., Kelsey, R., Clausen, T., and T. Winter, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks", draft-ietf-roll-rpl-19 (work in progress), March 2011.

[I-D.moskowitz-hip-rg-dex]
Moskowitz, R., "HIP Diet EXchange (DEX)", draft-moskowitz-hip-rg-dex-05 (work in progress), March 2011.

[I-D.ohba-pana-keywrap]
Cragie, R., Duffy, P., Ohba, Y., and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA) Extension for Key Wrap", draft-ohba-pana-keywrap-04 (work in progress), September 2011.

[I-D.ohba-pana-relay]
Duffy, P., Chakrabarti, S., Cragie, R., Ohba, Y., and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA) Relay Element", draft-ohba-pana-relay-03 (work in progress), February 2011.

[NISTIR7628VOL1]
The Smart Grid Interoperability Panel - Cyber Security Working Group, "Guidelines for Smart Grid Cyber Security: Vol. 1, Smart Grid Cyber Security Strategy, Architecture, and High-Level Requirements", NISTIR 7628, vol. 1, 2010.

[RFC3547] Baugher, M., Weis, B., Hardjono, T., and H. Harney, "The Group Domain of Interpretation", RFC 3547, July 2003.

[RFC3830] Arkko, J., Carrara, E., Lindholm, F., Naslund, M., and K. Norrman, "MIKEY: Multimedia Internet KEYing", RFC 3830, August 2004.

[RFC3971] Arkko, J., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, March 2005.

[RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, March 2005.

[RFC4279] Eronen, P. and H. Tschofenig, "Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)", RFC 4279, December 2005.

[RFC4347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security", RFC 4347, April 2006.

[RFC4423] Moskowitz, R. and P. Nikander, "Host Identity Protocol (HIP) Architecture", RFC 4423, May 2006.

[RFC4535] Harney, H., Meth, U., Colegrove, A., and G. Gross, "GSAKMP: Group Secure Association Key Management Protocol", RFC 4535, June 2006.

[RFC4962] Housley, R. and B. Aboba, "Guidance for Authentication, Authorization, and Accounting (AAA) Key Management", BCP 132, RFC 4962, July 2007.

[RFC5204] Laganier, J. and L. Eggert, "Host Identity Protocol (HIP) Rendezvous Extension", RFC 5204, April 2008.

[RFC5247] Aboba, B., Simon, D., and P. Eronen, "Extensible Authentication Protocol (EAP) Key Management Framework", RFC 5247, August 2008.

[RFC5281] Funk, P. and S. Blake-Wilson, "Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0)", RFC 5281, August 2008.

[RFC5295] Salowey, J., Dondeti, L., Narayanan, V., and M. Nakhjiri, "Specification for the Derivation of Root Keys from an Extended Master Session Key (EMSK)", RFC 5295, August 2008.

[RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

Appendix A. Examples of Node Configuration

Before any detail on methods is explored, the following section will provide various examples this document could cover. Exact requirements will be brought forward in subsequent sections. For the reader's general understanding this section is placed to give an idea of an acceptable usage scenario.

A.1. Smart Energy

A.1.1. Initial Meter Installation

The meter is initially loaded with code and network keys through a physical interface at the factory. The meter is installed at a customers home, and configured by the installer through the backbone link (via GSM modem, etc). Both operations can be performed through methods defined herein.

A.1.2. Home Expansions

The user wishes to join a thermostat onto the network. They press a button on the thermostat, which enters join mode. They press a button on the smart meter, which allows nodes to join the network. The devices both have displays, so they display a certain number which the user verifies match on both devices. The thermostat has now securely joined the network.

A.2. Consumer Products

A.2.1. Connecting DVD Remote to DVD Player

The user pushes a join button on the DVD remote and DVD player. The devices find each other, and blink in unison to indicate to the user which two devices will join. The user presses the button to confirm this, and the two devices are now joined together.

A.2.2. Adding a TV to a network with a DVD player and remote

The user then presses the join button on the DVD player and TV. The devices again find each other and blink in unison, with the addition that the remote control also blinks to indicate it is present in the network.

A.2.3. Providing GPS Location Data

A user has a simple GPS receiver (that has no user interface) they wish to broadcast location data with. The user switches on their camera, and enters a PIN from the base of the GPS. The user can now view GPS information such as satellite health from their camera. In addition photos are automatically tagged with location information.

A.3. Commercial Building Automation

A.3.1. Light Installation

The electrician installs the light fixture. Each light has a barcode printed on it. They use a handheld barcode scanner tool, which acts as the commissioning tool. When they scan a barcode with the tool, the tool asks the electrician to enter some additional information such as light fixture location. The tool securely registers the light fixture on the network, along with setting parameters inside the light fixture.

Appendix B. Example Exchanges

The following details how the protocol handles certain conditions and situations through examples. Note that each example is a more detailed description of the examples in Appendix A.

B.1. Smart Energy: Meter Manufacture

B.2. Smart Energy: Meter Installation

B.3. Smart Energy: Home Expansion

B.4. Consumer: Connecting DVD Remote to DVD Player

Supported User Interface Profiles

Profile	DVD Player	Remote Control
none	Y	Y
simple	Y	Y
numerical	Y	N
alphanumerical	Y	N
Graphical	Y	N

Supported Bootstrap Transport Layers

Layer	DVD Player	Remote Control
Physical	Y	Y
802.15.4	Y	Y
IrDA	Y	N

Supported Security Methods

Method	DVD Player	Remote Control
None	Y	Y
EAP	Y	N
Asymmetric, User	Y	Y
Asymmetric, CA	Y	N

The DVD player and remote control have a number of ways in which they could be joined together. The remote control does not have any unique identifier printed on it, thus no pre-shared key can be identified. This leaves either an unsecure joining method, or some asymmetric security method.

The remote control has a button on it for 'join', as does the DVD player. The user pushes the button on the DVD player, and then pushes the button on the remote control. Based on the UI profile, this causes the following to occur:

- DVD Player scans for existing network in advertise mode. Finding none, it starts a new network and that network enters advertise mode.

- The DVD remote scans for a network, and then finds the DVD player's network.

- The devices generate a shared secret (ie: Diffie-Hellman), and both blink their LED in a unique pattern based on this shared secret.

- The user user confirms both devices are blinking the same pattern, as both LEDs are blinking in unison.

- The DVD player displays 'JOIN OK' on it's LCD panel.

B.5. Consumer: Adding a TV to a network with a DVD player and remote

This network will have three devices: a TV, a DVD Player, and a Remote Control. The user will run the bootstrap protocol between the TV and Remote Control in this example, although it could also be run between the TV and DVD player.

Supported User Interface Profiles

Profile	TV	Remote Control
none	Y	Y
simple	Y	Y
numerical	Y	N
alphanumerical	Y	N
Graphical	Y	N

Supported Bootstrap Transport Layers

Layer	TV	Remote Control
Physical	Y	Y
802.15.4	Y	Y
IrDA	Y	N

Supported Security Methods

Method	TV	Remote Control
None	Y	Y
EAP-GPSK	Y	N
Asymmetric, User	Y	Y
Asymmetric, CA	Y	N

The TV and remote control have a number of ways in which they could be joined together. The remote control does not have any unique identifier printed on it, thus no pre-shared key can be identified. This leaves either an unsecure joining method, or some asymmetric security method.

The remote control has a button on it for 'join', as does the TV. In this example two sequence will be considered: where the TV button is pressed first, and where the remote control button is pressed first.

If the TV join button is pressed first:

- TV scans for existing networks in advertise mode. Finding none, it starts a new network and that network enters advertise mode.
- The remote scans for a network, and then finds the TV's network.
- The remote informs the TV it is on an existing network, and thus will require the TV to join this network.
- The devices generate a shared secret, and both blink their LED in a unique pattern.
- The DVD player in addition blinks, so the user is informed that if they confirm the join action the resulting network will have all three devices in it.
- The user confirms both devices are blinking the same pattern, as both LEDs are blinking in unison.
- The TV displays 'JOIN OK' onscreen, along with any information about the network it just joined.

If the remote control join button is pressed first:

- Remote control scans for existing networks in advertise mode. Finding none, it advertises it's network.

- The TV scans for a network, and then finds the remote control's network.
- The devices generate a shared secret, and both blink their LED in a unique pattern.
- The DVD player in addition blinks, so the user is informed that if they confirm the join action the resulting network will have all three devices in it.
- The user confirms both devices are blinking the same pattern, as both LEDs are blinking in unison.
- The TV displays 'JOIN OK' onscreen, along with any information about the network it just joined.

Appendix C. EAP, PANA, HIP-DEX and 802.1X

C.1. EAP Authentication Framework

EAP is an authentication protocol that supports a number of authentication algorithms called EAP methods [RFC5247]. EAP messages can be transported in different layers: using 802.1X, EAP messages are carried in Layer 2 and using PANA in IP or Layer 3 between EAP peer and the authenticator. EAP messages between the authenticator and authentication server are carried using AAA protocols (RADIUS or Diameter). The associated AAA server address of each resource-constrained device is assigned by the domain administrator. In the recommissioning case, another AAA server is reassigned to devices by the domain administrator if the device is switched to another domain.

At the end of a successful EAP method execution a master session key (MSK) is generated at both the EAP peer and EAP server. Authenticator receives MSK from EAP server at the end of EAP method execution using key transport. MSK is used in deriving a session key between the node and the authenticator using a protocol called secure association protocol (SAP). Derivation of the session key terminates bootstrapping of a node.

Additional keying material derived between EAP client and server that is exported by the EAP method is called Extended Master Session Key (EMSK). EMSK is not used in session key derivation but it could be used for the needs of other applications in higher layer protocols.

In the architecture introduced in Section 2.2 the node or router is the peer and the root is the authenticator. When the supplicant and authenticator are one hop away the authenticator can be reached

directly. However, this is rarely the case. In other cases the authenticator authenticates neighboring supplicants first. The router nodes that are authenticated become relay authenticators in the next phase and they relay authentication messages from the supplicants to the authenticator and vice versa. This continues until all nodes are authenticated.

EAP is a lock-step protocol, i.e. it executes in pairs of EAP-Request messages sent by the server and EAP-Response messages sent by the peer. At the end, the server indicates the status of authentication, usually by EAP-Success message which also carries the MSK. The first EAP-Request/Response pair is used for the server to request the identity and the peer to provide it. In the other pairs of EAP exchanges EAP method is executed.

Several EAP methods have been standardized each for different purposes. To authenticate devices with certificates, EAP Transport Layer Security (TLS) v1.2 specified in [RFC5216] which supports certificate-based mutual authentication is used.

Zigbee Alliance's Smart Energy Profile 2.0 Application Protocol Specification [SE2.0] mandates each device to be factory programmed with a certificate. The certificate is bound to a unique network ID, e.g. the device's MAC address or EUI-64 address. During EAP-Identity exchange the EAP peer provides its EUI-64 address as an identity to EAP server. This enables the server to validate the device certificate.

There are three bootstrapping scenarios using EAP.

1. Use of EAP for bootstrapping link-layer security.

   In this case, EAP is used for network access authentication to bootstrap link-layer ciphering. Security for higher-layer (i.e., IP layer and above) protocols is bootstrapped from an IB or OOB mechanism other than EAP.

2. Use of EAP for bootstrapping higher-layer security.

   In this case, EAP is used as an OOB mechanism for higher-layer authentication to bootstrap ciphering keys for one or more higher-layer protocols independently of network access authentication. When bootstrapping Constrained Application Protocol (CoAP) security with DTLS protection, a PSK (Pre-Shared Key) credential in the combined usage of DTLS (Datagram Transport Layer Security) [RFC4347] and PSK mode of TLS [RFC4279] is derived from EAP key material and DTLS ciphering keys are generated as a result of a successful DTLS handshake. Similarly,

when bootstrapping CoAP security with IPsec ESP protection, a PSK credential of IKEv2 [RFC5996] is derived from EAP key material and IPsec ESP ciphering keys are generated as a result of a successful IKEv2 handshake.

The ability to bootstrap multiple higher-layer protocols from a single execution of PANA authentication is important to save the computational resources for resource-constrained devices especially where public-key based authentication is used.

3. Use of EAP for bootstrapping both link-layer and higher-layer security.

   This case is the combination of the other two cases, and the most optimized way for bootstrapping resource-constrained devices. This case is only applicable where both the network access authentication and the higher-layer authentication use the same authentication server with the same authentication credentials.

The second and third scenarios are generally referred to as Single Sign-On in Section 4.2.2.2 of [NISTIR7628VOL1], where the root keys for higher-layer protocols can be derived from EAP EMSK (Extended Master Session Key) as an USRK (Usage-Specific Root Key) [RFC5295].

C.2. PANA

PANA (Protocol for carrying Authentication for Network Access) [RFC5191] defines an EAP transport over UDP where a PANA Client (PaC) is an EAP peer and a PANA Authentication Agent (PAA) is an EAP authenticator.

PANA can achieve the authentication, key freshness and data confidentiality objectives of security bootstrapping.

Multi domain operation is intrinsically supported due to the use of EAP and AAA.

Even though PANA architecture consisting of PaC, PAA and AAA Server is generic enough to be used in security bootstrapping, the architecture introduced in Section 2.2 requires a new element called PANA Relay Element (PRE). PRE is needed to enable PANA messaging between a PaC and PAA because the two nodes cannot reach each other by means of regular IP routing since only a link-local IPv6 address can be used by PaC prior to the completion of a succesful authentication.

PRE which is one hop away from PaC receives PANA messages and relays the message contents (payload) by encapsulating it in a message

parameter called Attribute Value Pair (AVP). PRE also needs to send header contents such as PaC's IP address and UDP port number in a different AVP. PRE has IP routing established with PAA which could be several hops away. PAA sends its reply messages in which the payload is encapsulated in an AVP. It also adds the AVP containing PaC's IP address and UDP port number. PRE creates a link local PDU using these AVPs and sends it to PaC [I-D.ohba-pana-relay].

The requirements for the use of PANA as a bootsrapping protocol can be stated as follows:

- A new entity called PANA Relay Element may be added to the PANA architecture. Behaviour of PANA Relay Element needs to be defined. New AVPs needed for PANA Relay Element operation for properly relaying messages from the client to the authenticator and vice versa are required to be specified.

- An extension to PANA to securely distribute keys from the PANA Authentication Agent to the PANA Client using AES Key Wrap with Padding algorithm needs to be defined. This is needed in order to use PANA for multicast group key distribution.

C.3. HIP-DEX

[RFC4423] introduces the Host Identity Protocol (HIP) where the Host Identity (HI) is a Cryptographic key (RSA, DSA, or ECC). A 128-bit length Host Identity Tag (HIT) is derived from the HI (hashed) and functions as an IPv6 address (/128 prefix) for applications. A four-packet Peer-to-Peer Host Identity Protocol Base EXchange (HIP BEX) establishes a security association (SA, similar to IKE), indexed by the HITs, but independent of the IP address. So HIP can be considered as a shim layer between the transport(TCP/UDP) and IP, providing authentication, data confidentiality, mobility in one basket.

As with IKE, HIP is typically used as a Key Management Protocol (KMP) for ESP. However, HIP is independent of IP and ESP and can be used for a KMP for any secure communication protocol at any level. Thus HIP can provide keying material for the MAC, IP, and Transport layers. HIP can be run directly over the MAC in an Ethernet Frame or within an Information Element in a MAC control frame. HIP can be run over UDP, traversing firewalls, and push keys over to Transport security protocols like SRTP or (D)TLS. Further, HIP could start on the MAC, then be enveloped over UDP after the first link.

The HIP-BEX involves many crypto primitives that are difficult to run on constrained nodes. HIP Diet Exchange (HIP-DEX) [I-D.moskowitz-hip-rg-dex] is a way to make HIP lightweight.

Basically, HIP-DEX a variant of the HIP-BEX specifically designed to use as few crypto primitives as possible yet still provide the same class of security features as HIP-BEX.

HIP-DEX can be used for mutual authentication between two endpoints. After mutual authentication, the two endpints establish a shared secret, which is fresh and fed into the encryption algorithm for data confidentiality. So HIP-DEX can achieve the authentication, key freshness and data confidentiality objectives of security bootstrapping.

When a node wants to authenticate to the network using HIP and Diet-HIP, it should be able to complete the HIP-BEX or HIP-DEX with the trust anchor or some delegate. In HIP, it does not matter how many domains, and nodes can authenticate each other as long as they have the secret.

In the architecture introduced in Section 2.2 the node and router could be the HIP end-points. Or the router could be a HIP Rendezvous Server, with the node registering to the rendezvous server (RVS) [RFC5204] to be reachable by other nodes. Typically the initial interaction will have the node be the HIP DEX Initiator with the router being the Responder. Using the RVS function on a Router any node could be the Initiator with any other Responder node. As long as there is IP connectivity between the Initiator and Responder, they can be multiple hops away from each other. Alternatively if the first hop from the Initiator has knowledge of the IP address of the Responder, it can act as a MAC/IP gateway for HIP.

An important requirement for the HIP-DEX to work in the architecture, the initiator should be able to get the IP address of the responder or have a gateway function as a forwarder. The IP address can be obtained from a 3rd party source like DNS of Distributed Hash Table (DHT), from local configuration, or from an RVS.

C.4. 802.1X

IEEE 802.1X defines how EAP packets can be transported over in Layer 2, i.e. Ethernet frames [802.1x] by encapsulating EAP packets into EAP Over Lan (EAPOL) frames between EAP peer, called supplicant and the authenticator. EAPOL can also be used in 802.11 wireless links.

To enable IEEE 802.15.4 devices to use EAP authentication, EAP packets encapsulated in EAPOL frames can be carried as payload in 802.15.4 data frames [802.15.4]. EAPOL is well defined and widely used and it lends itself to be easily carried in 802.15.4 data frames. For this, Frame Type subfield of the Frame Control Field of IEEE 802.15.4 MAC header needs to be set to a special value to

indicate the type of the payload, i.e. 802.1X encapsulated EAP packets. EAPOL packets are encoded following common EAPOL PDU structure defined in [802.1x] into the data payload field of 802.15.4 data frames.

Authentication proceeds as follows: authenticator authenticates the supplicants that are on the next hop first. This enables a secure link between the authenticator and these first-hop nodes. The architecture introduced in Section 2.2 requires a new entity called Relay Authenticator. First-hop nodes or router become Relay Authenticators in the next phase of authentication. Relay Authenticators tunnel EAPOL frames to the authenticator in the secure link established. This way all the supplicants are gradually authenticated.

After the keys are established from a successful EAP method (such as PSK mode of TLS), the node runs neighbor discovery protocol to get an IPv6 address assigned [I-D.ietf-6lowpan-nd]. Data transfer can be secured using DTLS or IPSec. Keys derived from EAP TLS are used in either generating DTLS ciphering keys after a successful DTLS handshake or IPSec ESP ciphering keys after a successful IKEv2 handshake.

802.1X can achieve the authentication, key freshness and data confidentiality objectives of security bootstrapping.

Multi domain operation is intrinsically supported due to the use of EAP and AAA. In order to support a device recommissioning case whereby the device's administrative domain is modified, after a new AAA server address assigned and a successful 802.1X method execution, the old set of device 'name and password' MUST be overwritten into the device by a new set of 'name and password' that are assigned by the domain administrator. The device MUST be rebooted to execute EAP method again for authentication and a new master session key MUST be generated.

It should be noted that currently 802.15.4 does not allow multiplexing multiple protocols on the same link like ethernet does. However this might change in the future.

The requirements for the use of 802.1X defined EAPOL as a bootsrapping protocol can be stated as follows:

- A special value in the Frame Type subfield of the Frame Control Field of IEEE 802.15.4 MAC header to indicate the type of the payload,

- Link Layer Multicast Group addresses for 802.15.4 corresponding to EAPOL Group Address Assignments defined in Table 11.1 of [802.1x], especially to be used in EAPOL-Start packet.

- Which MAC frames of beacon, data, acknowledgment and MAC command as defined in [802.15.4] with what security levels are mapped to controlled port, which MAC frames with what security levels are mapped to uncontrolled port and which MAC frames are never mapped to any of controlled/uncontrolled port (i.e., the payload of those frames are used by the MAC-layer ieself and never used by upper layers).

- A new entity called Relay Authenticator may be added to the 802.1x architecture. Behaviour of Relay Authenticator needs to be defined.

Authors' Addresses

Behcet Sarikaya
Huawei USA
1700 Alma Dr. Suite 500
Plano, TX 75075

Email: sarikaya@ieee.org

Yoshihiro Ohba
Toshiba
Tokyo, Japan

Email: yoshihiro.ohba@toshiba.co.jp

Robert Moskowitz
Verizon Business Systems
15210 Sutherland
Oak Park, MI 48237

Email: rgm@labs.htt-consult.com

Zhen Cao
China Mobile
Beijing, China

Email: caozhen@chinamobile.com

Robert Cragie
Pacific Gas and Electric
89 Greenfield Crescent
Wakefield, UK WF4 4WA

Email: robert.cragie@gridmerge.com