ace
Internet-Draft
Intended status: Standards Track
Expires: June 14, 2014

B. Sarikaya
Huawei USA

December 11, 2013

Secure Bootstrapping Solution for Constrained Devices
draft-sarikaya-ace-secure-bootstrapping-00

Abstract

We present a solution to initially configure the network of resource constrained nodes securely, a.k.a., secure bootstrapping. The solution is based on EAP-TLS authentication with the use of raw public keys as certificates.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

Bootstrapping is any processing required before the network can operate. 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.

Bootstrapping needs to be secure to make sure that the network operation is secure and hence secure bootstrapping ensures that only the authorized nodes can get access to the network. Because of this secure bootstrapping needs to preceed IP address configuration.

[I-D.jennings-core-transitive-trust-enrollment] defines a protocol that enables sensors to securely connect into a system that uses them. The protocol which is being defined is based on the Device using HTTP or COAP [I-D.ietf-core-coap] to communicate with the Controller. This seems to assume that the device is already configured with an IP address. Such an assumption violates the assumption we have in this document on secure bootstrapping.

Transport Layer Security (TLS) is commonly used protocol to secure web browsing, emailing, or other client-server applications. In TLS, the client and the server present their certificates and authenticate each other. Recently, raw public key extension is defined to be used as certificates [I-D.ietf-tls-oob-pubkey]. In this document we use the raw public keys in EAP-TLS.

The document continues in Section 2 on bootstrapping architecture, in Section 3 on secure bootstrapping solution, in Section 4 on transporting EAP messages, in Section 5 on future work.

2. Secure Bootstrapping 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. [RFC6550] 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 3 describes the bootstrapping procedures where EAP (Extensible Authentication Protocol) [RFC3748] and other protocols are used as the bootstrapping protocols.

3. 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 [RFC6775]. 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 [RFC5247].

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 by sending an EAP-Request/Identity packet. The device replies with an EAP-Response/Identity containing the device's ID. 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-TLS/Start packet which is an EAP-Request packet with EAP-Type=EAP-TLS 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. EAP-Response packet MUST have the EAP-Type value set at EAP-TLS Figure 1.

The device MUST add an extension of type client certificate type and server certificate type defined in [I-D.ietf-tls-oob-pubkey] to Client-Hello message. Both of these types 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, server certificate type set to RawPublicKey and includes only the SubjectPublicKeyInfo part of the certificate with its raw public key, rather than the whole certificate in the Certificate message and then sends them to the device with an EAP-Request/TLS-Server-Hello message. Server-Key-Exchange message contains a temporary key for the client to encrypt Client Key Exchange message. For the device, the server adds certificate request message to ask for the device's RawPublicKey using client certificate type message.

Device receives AS's RawPublicKey. Device SHOULD verify the key using out of band mechanisms. Device sends Client Certificate message containing the device's RawPublicKey. 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 encrypting pre_master_secret with the temporary key in Server-Key-Exchange message. Device includes Change Cypher Spec message to indicate that all messages that follow Client Finished message will be encrypted.

The AS derives the Master Session Key (MSK) and replies with EAP-Request/TLS-Server-Finished message. In this message, the server includes Change Cypher Spec message to indicate that the server will beging encrypting messages with the keys negotiated. The 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 device. The AS MUST send the group key in this message and the EAP-TLS ends.

```
     Device             Authenticator       Authentication
      |                                          Server
      Device connects to      |                       |
      Network                 |                       |
      |<----TLS-Start---------|<---EAP-Request--------|
      |-TLS-ClientHello------>|----EAP-Response------>|
      |client certificate type|                       |
      |server certificate type|                       |
      |<-TLS Server Hello-----|<----EAP-Request-------|
      |                       |server certificate type|
      |                       | Server Certificate    |
      |                       |client certificate type|
      |                       | Certificate Request   |
      |                       | Server Key Exchange   |
      |                       | Server Hello Done     |
      |TLS-ClientCertificate->|----EAP-Response------>|
      |Client Key Exchange    |                       |
      |Change Cipher Spec     |                       |
      |Client Finished        |                       |
      |<TLS Change Cipher Spec|<----EAP-Request-------|
      |                       | Server Finished       |
      |TLS null-------------->|----EAP-Response------>|
      |      EAP-Success      |<----EAP-Success-------|
      |Authentication finished|                       |
```

Figure 1: Authentication Call Flow

4. Transporting EAP Messages

EAP can be transported between the device and the authenticator either in Layer 3 using PANA [RFC5191] or in Layer 2 using IEEE 802.1X [802.1x].

EAP is transported using RADIUS [RFC2865] between the authenticator and authentication server.

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, such as the PANA Relay Element (PRE).

After the keys are established from a successful EAP method (such as EAP-TLS), the device runs neighbor discovery protocol to get an IPv6 address assigned [RFC6775]. 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.

5. Future Work

The nodes in a constrained network called devices have wide range of capabilities and are used in diverse number of applications. Different secure bootstrapping solutions may apply to different applications and different types of nodes. In all cases, it is assumed in this document that the devices are IPv6 enabled.

The solution described in Section 3 has the most stringent requirements on the devices and therefore is not suitable on less constrained nodes. It seems that the devices used in smart metering may have enough resources to run the bootstrapping protocol and they do not suffer from power constraints compared with most other devices such as light switches.

One possible optimization in Figure 1 applies to the case where the device does not have a RawPublicKey. In this case the device sends only server_certificate_type set to RawPublicKey in Client-Hello message. In response, AS sends its RawPublicKey in Server Hello message. As a result the messages are much simpler than in Figure 1.

Further optimizations to the EAP-TLS call flow in Figure 1 are TBD.

Simpler devices such as light switches, environmental sensors, etc. may have much less resources, much less constrained IPv6 stack and they may not stay on for long periods of times required from the execution of the secure bootstrapping protocol.

Identification of a set of applications with similar device capabilities is TBD.

Modification of the protocol defined in Section 3 to define a secure bootstrapping protocol for each set is TBD.

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

TBD.

9. Acknowledgements

TBD.

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.

[I-D.ietf-tls-oob-pubkey]
Wouters, P., Tschofenig, H., Gilmore, J., Weiler, S., and T. Kivinen, "Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", draft-ietf-tls-oob-pubkey-10 (work in progress), October 2013.

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

[RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)",

RFC 2865, June 2000.

[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.

10.2. Informative References

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

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

[I-D.ietf-core-coap]
Shelby, Z., Hartke, K., and C. Bormann, "Constrained Application Protocol (CoAP)", draft-ietf-core-coap-18 (work in progress), June 2013.

[I-D.jennings-core-transitive-trust-enrollment]
Jennings, C., "Transitive Trust Enrollment for Constrained Devices", draft-jennings-core-transitive-trust-enrollment-01 (work in progress), October 2012.

[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.

[RFC6550] Winter, T., Thubert, P., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, March 2012.

[RFC6775] Shelby, Z., Chakrabarti, S., Nordmark, E., and C. Bormann, "Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 6775, November 2012.

[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.

Author's Address

Behcet Sarikaya
Huawei USA
5340 Legacy Dr.
Plano, TX 75024

Email: sarikaya@ieee.org