INTERPRETATION IC 135-2008-15 OF
ANSI/ASHRAE STANDARD 135-2008 BACnet® -
A Data Communication Protocol for Building
Automation and Control Networks

Approval Date: January 29, 2011

Request from: Dean Matsen (dean.matsen@honeywell.com), Alerton Dealer Business Honeywell Automation & Control Solutions, 6670 185th Ave. NE, Redmond, WA 98052.

Reference: This request for interpretation refers to the requirements presented in Addendum g to ANSI/ASHRAE Standard 135-2008, Clauses 24.2.3 and 24.12.1 (pages 11 and 40), relating to Selecting General-Network-Access key for encryption when another key is used for signing.

Background: Clause 24.2.3 states "If the do-not-decrypt flag has a value of 0, the General-Network-Access key is used to decrypt the message as it is the only key that is guaranteed to be known by intermediate routers (see Clause 24.21.1)."

Clause 24.12.1 states "A signed message shall be encrypted using the General-Network-Access key if the security policy of the outgoing port is 'encrypted'. The do-not-decrypt flag shall be set to 0."

Both of these paragraphs allude to the possibility that the key identifier in the security header may only indicate the key used to sign the message, not the one used to encrypt/decrypt it (and in cases where the keys are different, it is the General-Network-Access key that is used to encrypt/decrypt).

Both of these paragraphs also use the term "the General-Network-Access key" as if there is only one. In fact, there can be two General-Network-Access keys active at one time during the overlapping period between two key sets. There is no guideline given for how to choose which GNA key to use during this time.

Failure to specify this is very likely to lead to interoperability failures where one device uses one algorithm to choose the GNA key for encryption, and another devices uses a different algorithm to choose the GNA key for decryption. The algorithm for choosing the GNA key in this situation needs to be the same for both the sending and the receiving device.

Furthermore, there is no information given on what error should be returned if a device receives a message that needs to be decrypted with the GNA, but the device does not have a matching GNA key with which to decrypt it. On the surface, it seems that returning "unknownKey" or "unknownKeyRevision" might be appropriate, but there are three problems with doing this:

1. "unknownKey" requires specifying the key identifier of the key that was unknown. Normally, this is no problem because this information is given in the security header, but in this case, the security header could be specifying some other key/algorithm/revision. Therefore, the responding device could fill in the key number "General-Network-Access", but it does not have any way to fill in the algorithm.

2. "unknownKeyRevision" requires specifying the revision that was unknown. This poses the same problem as item 1, above.

3. Above all, since the incoming message was received encrypted with the GNA key, such a security response would also have to be encrypted using the GNA key, and since there is an apparent mismatch in GNA keys between the two devices, the other device would not be able to decrypt the response anyway.

Interpretation No.1: If the security header indicates the General-Network-Access key, then the device should use the algorithm and version of GNA key specified in the header for both signing and for encryption -- there is no need to use a separate algorithm to find the GNA key.

Question No.1: Is this interpretation correct?

Answer No.1: Yes.

Interpretation No.2: If the security header indicates a key other than General-Network-Access, and the message needs to be encrypted/decrypted, then the device should use the most recent General-Network-Access key that is within its active period based on the timestamp in the message (NOT the current timestamp).

Question No.2: Is this interpretation correct?

Answer No.2: No (the standard is clear that the security header indicates the key revision to use).

Interpretation No.3: If a receiving device cannot find a usable General-Network-Access key to attempt to decrypt the message, then the message should be dropped.

Question No.3: Is this interpretation correct?

Answer No.3: No (the unknownKeyRevision error handling is applicable to this case).

Comments: See answers.