Amazon SNS Release Note – April 26, 2010

Details on Computing a Notification Message Signature

The first step is to extract the needed fields of the message from the JSON format. One should use a well-behaved JSON parser to extract the fields needed below. It is important to use a JSON parser as some control characters are escaped (such as newlines converted to '\n') while the text was converted to JSON. In extracting the text the reverse conversion must be done.

The notification message signature is computed using the SHA1withRSA algorithm on a "canonical string" - a UTF-8 string constructed via the following function:

```
StringBuilder sb = new StringBuilder();
sb.append("Message\n");
sb.append(message).append("\n");
sb.append("MessageId\n");
sb.append(messageId).append("\n");
if (subject != null) {
   sb.append("Subject\n");
   sb.append(subject).append("\n");
}
sb.append("Timestamp\n");
sb.append(timestamp).append("\n");
sb.append("TopicArn\n");
sb.append(topicarn).append("\n");
sb.append("Type\n");
sb.append(type).append("\n");
return sb.toString().getBytes("UTF-8");
```

To verify the signature, first get the notifications service certificate (e.g. http://sns.us-east-1.amazonaws.com/SimpleNotificationService.pem). Extract the public key, decode the base64-encoded signature, and use a tool like openssl to verify the signature cryptographically. For example, given files CERT, SIG and MESS which contain the certificate, signature and the canonical string:

```
> openssl x509 -in CERT -pubkey -noout > pub
> base64 -i -d SIG > sigraw
> openssl dgst -sha1 -verify pub -signature sigraw MESS
```

Details on Computing a Subscribe or Unsubscribe Message Signature

The first step is to extract the needed fields of the message from the JSON format. One should use a well-behaved JSON parser to extract the fields needed below. It is important to use a JSON parser as some control characters are escaped (such as newlines converted to '\n') while the text was converted to JSON. In extracting the text the reverse conversion must be done.

The subscribe message signature is computed using the SHA1withRSA algorithm on a "canonical string" - a UTF-8 string constructed via the following function:

```
StringBuilder sb = new StringBuilder();
sb.append("Message\n");
sb.append(message).append("\n");
sb.append("MessageId\n");
sb.append(messageId).append("\n");
sb.append("SubscribeURL\n");
sb.append(subscribeurl).append("\n");
sb.append("Timestamp\n");
sb.append(timestamp).append("\n");
sb.append("Token\n");
sb.append(Token).append("\n");
sb.append("TopicArn\n");
```

```
sb.append(topicarn).append("\n");
sb.append("Type\n");
sb.append(type).append("\n");
return sb.toString().getBytes("UTF-8");
```

The signature can be verified using an Amazon public key, as described in the documentation on the notification message format above.