Device Pairing: Lessons Learned

Cary Bran and Joseph Stachula

Smart, sensor-oriented networks are steadily marching towards mainstream adoption in the marketplace. As smart object adoption increases so will the demand that traditional wearable devices such as mobile phones and headsets interoperate and participate in the smart object ecosystem. To start down the path of the interoperability discussion we would propose that the workshop explore some of the lessons learned from the Bluetooth headset market. In particular we would propose that some time be given in this workshop explore what a standard mechanism for secure pairing and data transport would look like.

This paper is intended to share a few of the pairing issues we have seen as a market leading headset manufacturer. The issues discussed here are not specific to any one vendor and have been made as generic as possible for sharing with the group.

Pairing Devices
When Bluetooth devices initially pair, a random numeric pairing code is used to generate a unique security token that is shared between the devices. The security token is then stored and exchanged in all subsequent transactions. This is done to avoid repeating the pairing process when devices that were previously paired reconnect. While the pairing mechanism looks good on paper, a problem emerges when the device’s user interface is minimized or altogether removed.

With little or no user interface, devices that use a randomly generated pairing code become very cumbersome as there is no way to relay the code to the user. Many device vendors work around this problem by using a known pairing code such as “0000”. The “0000” approach works for users/environments where secure device pairing is not important, but it also leaves device open for anonymous pairing which may not be desired.

To address the lack of user interface while at the same time keeping pairing codes random, headset vendors have developed headsets that leverage text to speech and automatic speech recognition for pairing code generation. The audio interface approach is compute-heavy and does not scale down well to power constrained environments such as sensor networks.

Assuming that human wearable devices such as mobile phones and headsets will be interoperable with smart objects work will need to be done to explore what can be improved or changed with current pairing approaches.
Securing Channels of Communication

When headsets pair, the overall security of the communication channels is only as secure the least secure device. For example if there are two devices: A and B. Assume that device A supports 32-bit encryption and device B supports 32 and 128-bit encryption. When device A pairs with device B the cipher strength of their communications channel will be the lowest common denominator, in this case is 32-bit encryption.

Note: If A and B do not agree on an encryption mechanism, the pairing will fail and the user will receive an error.

If the A and B do agree on an encryption mechanism the user has no way of knowing which cipher-strength they are using.

The case of negotiating encryption for pairing A to B is a trivial example, especially when compared to the complexities of pairing A or B with an ecosystem of smart objects. Lest assume a second simple pairing problem with four devices A, B, C and D. Devices A and B support data transmission with 32, 64 and 128-bit encryption and device C only supports 32-bit encryption, device D only supports 256-bit encryption. When A, B, C and D try to pair, only A,B and C will pair successfully. Device D will fail to pair, because it cannot support the least common encryption level in this case 32-bit.

When the above scenario happens with pairing Bluetooth devices, it results in a connection a connection error. This is a point of frustration with users and could be resolved if there was a standard security profile. To address this problem for smart objects we would propose a security baseline for all devices participating in a smart ecosystem.

Human Middleware

One could argue after reading this that Bluetooth has demonstrable pairing shortcomings. There may be some truth to this argument, but could there be another factor that is leading to the issues with pairing, the human factor. Bluetooth pairing mechanisms require human interaction in order to succeed. This begs the question: if human wearable devices are participants within a smart object world, could or should the human role in the pairing process be significantly reduced or removed all together? Ideally we would like to see smart object pairing smart and secure enough to allow users to confidently participate in a smart object ecosystem using no more than a binary decision (yes/no).