The Share-Send Algorithm (SSA) for Securing the Internet-of-Things

The Internet-of-Things (IoT) is at the centre of the emerging smart world, offering a level of automation in commerce, industry and other fields never seen before. However, security has failed to keep pace with the technology, and is in many cases non-existent.

The Share-Send Algorithm (SSA) is our proposed solution to IoT security. It offers much greater security during transmission than other methods, but is computationally efficient and naturally allows multiple devices to collaborate in securely sending signals to each other.

What is the problem with implementing IoT security?

Implementing complex cybersecurity ciphers, such as RSA or AES, in computers is not a problem, as they have more than enough computing power to run them, but microchips, used in IoT devices, mostly do not have this luxury.

Many users therefore sacrifice security in favour of speed and battery life.

What are the security risks?

- **Data encryption during transmission**: It is one thing to protect a device, but another to protect sensitive information during transmission, in case a hacker intercepts it.
- **Data authentication**: Even if the data is encrypted, there is no guarantee it was sent by the expected source. Consider the brakes in a car being told to activate, or the heating in a room being instructed to switch off.
- **Side-channel attacks**: It has been shown that even the most advanced algorithms, such as 4096 bit RSA, can be cracked with such an attack, whereby a hacker can for example analyse the power consumption during encryption and use it to determine the secret key.

How can SSA help?

Data encryption during transmission: In SSA, only purely random numbers are ever transmitted. Even if the same signal is transmitted twice, two randomly different sequences of random numbers are sent.

Data authentication: SSA is based on secret sharing, which guarantees that only those holding the correct shares will be able to send and receive messages.

Side-channel attacks: SSA does not use cryptographic keys, in the conventional sense, so obtaining a key to unlock future messages is not possible.

In addition: SSA uses only simply binary operations, so is **computationally efficient, not draining batteries** as much as conventional ciphers. It is also a **collaborative form of cryptography**, allowing multiple devices or microchips to send a signal or message to multiple receivers. In this, a signal is only received if all the correct shares have been received. Finally, SSA can be rendered **time-sensitive**, ensuring that a signal or message decoded at one time cannot be decoded at a later time.

What about other uses of SSA?

As well as in IoT, it can be used as a light-weight, highly secure stand-alone cipher for securing any information sent through websites, such as Gmail and social media. It can also be used to secure data in databases, so that even a fully compromised database only reveals random numbers to a hacker.