

UDPi August FAQs

1) Does UDPi use an “on/off” term from a crypto perspective? Also, does the polynomial term drop to 0 by multiplying by the CTR position?

No, UDPi does not simply use an “on/off” term. With UDPi, the term can be 0. Many algorithms do not allow a term of 0 to be valid, but with UDPi a term of 0 is acceptable. The polynomial term can be 0 and still be a valid term.

2) Why does UDPi’s use of randomness not make code easier to crack?

There is no real randomness in a computer, but rather it must be pseudo-random. The beauty of UDPi is that the flow of firmware in the external memory is unknown. The flow could be a result of a jump, GOTO instruction or simply normal flow, but UDPi makes the execution of flow from one location to another in a random manner. Some of the external addresses are accessed in the same order and some will differ according to an action or status in the processor.

3) Does UDPi use fuses, which can be seen with Xray and scanning?

Security fuses can be easily traceable with the use of certain technologies (which can be expensive). The best implementation of UDPi is to be programmed at the layering of the die by the laser (optics that expose the die to a light that will cause it to be etched or protected when exposed to a chemical). With this implementation of UDPi, Xray and scanning attacks are useless.

4) Is UDPi valuable in securing off-chip code? Is the protection worth the extra cost?

UDPi is valuable for securing off-chip code. It is simple to implement and cost effective. Protection and cost are frequently at odds with each other, but with UDPi both large and small processors can benefit from advanced protection at a low cost.

5) Is UDPi based on a formula, and are hash schemes and keys effective?

Many solutions based on a formula can be broken, but UDPi is not based on a formula. Hash schemes and keys are only effective if there is not a predictable formula used.

6) Is the encoding for UDPi strong?

If the code is broken, the attacker may have the IP, but with UDPi they cannot do anything with it. Even with the IP, with UDPi the attacker doesn’t have the clean text description of the IP. Also, two codes would have to be broken, including the CPU manufacturer’s code as well as the product’s code.

7) How does UDPI improve on anti-FIB techniques and other security technologies used in the smartcard market?

Many technologies need to be open and must use authentication. If a device must be designed to communicate with many other devices, then the communication must be open and there must be a common method, algorithm or the like to communicate with the other devices. This is why UDPI's technology does not suit the Internet, email or other applications that require an open communication. UDPI creates a secure link between two devices.

8) What prevents UDPI's technology from being reverse engineered?

UDPI cannot be reverse engineered because it is not based on a formula and accepts 0x00 as a term.

9) What is the UDPI technology? How is it commercialized? How is it delivered and integrated?

UDPI is a protection method for internal and external intellectual property (IP). It is commercialized through co-designing the CPU with manufacturers and is integrated as part of the CPU code.