SECURE 5 CLICK

PID: MIKROE-3774
Weight: 18 g

Secure 5 Click includes the ATECC508A, a secure CryptoAuthentication™ device from Microchip, which is equipped with an EEPROM array which can be used for storing of up to 16 keys, certificates, consumption logging, security configurations and other types of secure data. The ATECC508A equipped on this click board™, supports the SWI interface with a flexible command set, that allows use in various security applications, including Network/IoT Node Endpoint Security, Secure Boot, Small Message Encryption, Key Generation for Software Download, Ecosystem control, Anti Counterfeiting and similar.

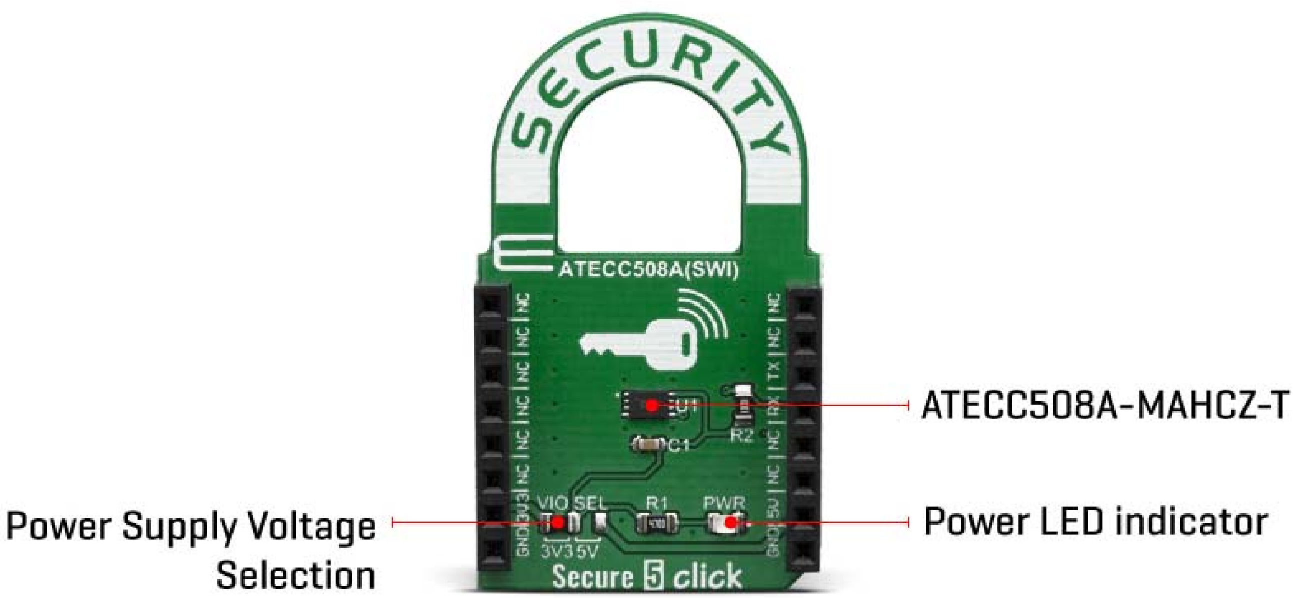
Secure 5 click board™ is supported by a mikroSDK compliant library, which includes functions that simplify software development. This Click board™ comes as a fully tested product, ready to be used on a system equipped with the mikroBUS™ socket.

The click board™ comes with stacking headers which allow you to combine it with other click boards™ more easily by using just one mikroBUS™ socket.

The EEPROM array that is included in the ATECC508A coprocessor can be used for storage of up to 16 keys, certificates, miscellaneous read/write, read-only or secret data, consumption logging, and security configurations. Access to the various sections of memory can be restricted in a variety of ways and then the configuration can be locked to prevent changes. Therefore, this Secure 5 click should mainly be used for security purposes.

HOW DOES IT WORK?

Microchip's ATECC508A integrates ECDH (Elliptic Curve Diffie Hellman) security protocol, an ultra-secure method to provide key agreement for encryption/decryption. It also integrates the ECDSA (Elliptic Curve Digital Signature Algorithm) sign-verify authentication for the Internet of Things (IoT) market, including home automation, industrial networking, accessory and consumable authentication, medical, mobile and more.



It features a wide array of defense mechanisms specifically designed to prevent physical attacks on the device itself, or logical attacks on the data transmitted between the device and the system. Hardware restrictions on the ways in which keys are used or generated provide further defense against certain styles of attack.

The ATECC508A has a flexible command set that allows use in many applications, including the Network/IoT Node Protection that authenticates node IDs, ensures the integrity of messages, and supports key agreement to create session keys for message encryption. It can also be used for Anti-Counterfeiting, meaning it validates that a removable, replaceable, or consumable client is authentic. Examples of clients could be system accessories, electronic daughter cards, or other spare parts. It can also be used to validate a software/firmware module or memory storage element. The next feature is Protecting Firmware or Media, which means it validates code stored in flash memory at boot to prevent unauthorized modifications, encrypt downloaded program files as a common broadcast, or uniquely encrypt code images to be usable on a single system only. Also, storing Secure Data, which means you can store secret keys for use by crypto accelerators in standard microprocessors. Programmable protection is available using encrypted/authenticated reads and writes. And finally, Checking User Password, and that ensures that it validates user-entered passwords without letting the expected value become known, maps memorable passwords to a random number, and securely exchanges password values with remote systems.

Access to the device is made through a standard SWI Interface at speeds of up to 1Mb/s, which can reduce the number of GPIOs required on the system processor, and/or reduce the number of pins on connectors. If the Single-Wire Interface is enabled, the remaining pin is available for use as a GPIO, an authenticated output or tamper input.

Each ATECC508A ships with a guaranteed unique 72-bit serial number. Using the cryptographic protocols supported by the device, a host system or remote server can verify a signature of the serial number to prove that the serial number is authentic and not a copy. Serial numbers are often stored in a standard Serial EEPROM; however, these can be easily copied with no way for the host to know if the serial number is authentic or if it is a clone.

The device is consuming very low current, especially while it is in the sleep mode. The chip itself uses less than 150nA, in that case. The voltage range which can be used to power up the Security 5 click, allows for it to work with both 3.3V and 5V capable MCUs. Therefore, this click board™ supports the parasitic power supply mode, where the main IC is powered via the communication line. When the onboard jumper PWR BYP is removed, Secure 5 click

The chip itself uses a minimal number of pins; only the SWI lines are routed to the mikroBUS™ along with the 3.3V and 5V rails. The device can work with any of these voltages. It can be selected by soldering a small SMD jumper, labeled as VIO SEL to the correct position.

IMPORTANT: On this click board™, UART lines (RX and TX) are shorted and pulled high by the 1KΩ resistor. Basicly, they act as a single line and only one trace is routed to the ATECC508A IC. Further it means that UART pins can be used only for SWI communication when this click board™ is used on a system.

SPECIFICATIONS

Type	Encryption
Applications	IoT node security and ID, secure download and boot, ecosystem control, message security, anti-cloning, etc.
On-board modules	ATECC508A cryptographic co-processor
Key Features	Performs high-speed public key (PKI) algorithms, NIST Standard P256 elliptic curve support, SHA-256 hash algorithm with HMAC option, 256-bit key length, storage for up to 16 Keys
Interface	SWI
Click board size	M (42.9 x 25.4 mm)
Input Voltage	3.3V or 5V

PINOUT DIAGRAM

This table shows how the pinout on Secure 5 click corresponds to the pinout on the mikroBUS™ socket (the latter shown in the two middle columns).

Notes	Pin	mikroBUS				Pin	Notes
	NC	1	AN	PWM	16	NC	
	NC	2	RST	INT	15	NC	
	NC	3	CS	RX	14	TX	SWI Line

	NC	4	SCK	TX	13	**RX**	SWI Line
	NC	5	MISO	SCL	12	NC	
	NC	6	MOSI	SDA	11	NC	
Power Supply	**3.3V**	7	3.3V	5V	10	**5V**	Power supply
Ground	**GND**	8	GND	GND	9	**GND**	Ground

ONBOARD SETTINGS AND INDICATORS

Label	Name	Default	Description
LD1	PWR LED	-	Power LED Indicator
JP1	PWR. SEL	Left	Power supply voltage selection, left position 3V3, right position 5V

SOFTWARE SUPPORT

We provide a library for the Secure 5 click on our LibStock page, as well as a demo application (example), developed using MikroElektronika compilers. The demo can run on all the main MikroElektronika development boards.

Library Description

The library demonstrates the operation of the software single wire interface implementation.

Key functions:

- `int8_t secureswi_init(T_SECURESWI_DIRSET inSet, T_SECURESWI_DIRSET outSet)` - Initialize the SWI interface and pass the pin direction setting functions.
- `void secureswi_sendBytes(uint8_t len,uint8_t *stBuf)` - Encode data buffer and send the data to the SWI bus.
- `void secureswi_receiveBytes(uint8_t len,uint8_t *stBuf)` - Receive and decode data from the SWI bus.

Examples description

The application is composed of three sections :

- System Initialization - Initialize the GPIO sturcture and configure the serial port for logging data.
- Application Initialization - Initialize the driver and configure swi for communication.
- Application Task - Data is read from the secure chip. If the readout is successful the data is then printed on the serial port in the hex format.

```
void applicationTask()
{
     uint8_t bufferOut[128];

     cfg_atsha204a_swi_default.iface_type  = ATCA_SWI_IFACE;
     cfg_atsha204a_swi_default.devtype     = ATSHA204A;
     cfg_atsha204a_swi_default.atcaswi.bus = 1;
     cfg_atsha204a_swi_default.wake_delay  = 2560;
     cfg_atsha204a_swi_default.rx_retries  = 10;

     atcab_init(&cfg_atsha204a_swi_default);

     mikrobus_logWrite("Starting test",_LOG_LINE);

     memset(bufferOut,0,127);

     if (atcab_read_serial_number(bufferOut) == ATCA_SUCCESS)
     {
         mikrobus_logWrite("rn Serial number: ",_LOG_LINE);
         secureswi_printHex(bufferOut,9);
     }
     else
     {
         mikrobus_logWrite("rn Reading serial number failed...",_LOG_LINE);
         secureswi_printHex(bufferOut,sizeof(bufferOut));
     }

     Delay_ms (1500);
     memset (bufferOut, 0x00, 128);
```

```
    if (atcab_read_config_zone(bufferOut) == ATCA_SUCCESS)
    {
       mikrobus_logWrite("rnrn First 32 bytes of device configuration: ",_LOG_LINE);
       secureswi_printHex(bufferOut,32);
    }
    else
    {
       mikrobus_logWrite("rnrn Reading config zone failed...",_LOG_LINE);
       secureswi_printHex(bufferOut,sizeof(bufferOut));
    }

    while(1)
    {

    }
}
```

The full application code, and ready to use projects can be found on our LibStock page.
Other mikroE Libraries used in the example:

- Conversions
- C_String
- UART

Additional notes and informations
Depending on the development board you are using, you may need USB UART click, USB UART 2 click or RS232 click to connect to your PC, for development systems with no UART to USB interface available on the board. The terminal available in all MikroElektronika compilers, or any other terminal application of your choice, can be used to read the message.

MIKROSDK

This Click board™ is supported with mikroSDK - MikroElektronika Software Development Kit. To ensure proper operation of mikroSDK compliant Click board™ demo applications, mikroSDK should be downloaded from the LibStock and installed for the compiler you are using.

For more information about mikroSDK, visit the official page.