THE ACCESS MANAGEMENT SYSTEM FOR THE PARASITE-POWERED DEVICES

D. JAROS L. MACHAN
Department of Microelectronics, FEEC, Brno University of Technology, Brno, Czech Republic
Technicka 3058/10, 616 00 Brno, Phone: +420541146180, Email: machan@feec.vutbr.cz, jarosd@feec.vutbr.cz

R. KUCHTA J. KADLEC
Department of Microelectronics, FEEC, Brno University of Technology, Brno, Czech Republic
Technicka 3058/10, 616 00 Brno, Phone: +420541146193, Email: kuchtar@feec.vutbr.cz, kadlecj@feec.vutbr.cz

Abstract: *This paper describes a possible solution for access management in low-power systems. This proposal could be used in the environments, where it is necessary for a secured service provided by devices that are parasitically powered. For communication a 1-Wire serial interface is used. From the abstract point of view, the system matches to AAA system (Authentication, Authorization and Accounting). The security is based on symmetric encryption algorithm AES-128 (original Rijandael).*

Key words: *AES-128, 1-Wire, AAA, authentication, low-power*

1. Introduction

This paper shows a possible solution for applications where non battery powered and secured access are required simultaneously. The top level point of view is depicted in the figure 1. The situation reflects general AAA system [1], [2], [3], [4] and [5]. There is an authority which delegates a system administrator to manage access rights for users, especially their identifications and setting encryption keys in our system. The user is an entity (electronic device) that wants to gain access to the secured service provided by the server. The server performs authentication. After the authentication of the user, the server provides authorization and grants or rejects access to the secured service. For our purpose the user acts also as power source for the server. Both of basic entities (user and server) are the logging of events. This log could be then read out by the system administrator. The result of a successful authentication and authorization process can be followed by an action. For example some type of actuator, like a miniature DC motor or solenoid which can be actuated.

In Section 2, there is a description of the circuit background and top level system diagram. The authentication scheme is described in Section 3. Section 4 is focused on specific parameters of FRAM memory used in MSP430 microcontrollers. Section 5 is concerned with the discussion of Rijandael encryption algorithm and its time effectivity. Future works and plans are mentioned in Section 6. In Section 7, there is a conclusion which summarizes the results of the work.

2. Circuit background

Both parties (client and server) communicate through 1-Wire serial interface [4] as depicted in the figure 2. The client works as master and the server works as slave. As mentioned previously the slave (server) is parasitically powered from the master (client). The electrical energy for powering the slave is stored in the capacitor C_{ST}. This capacitor has to be charged prior to the communication. On the master side you can see two pull-up resistors. The resistor R_{DPU} is used during slave detection and communication.

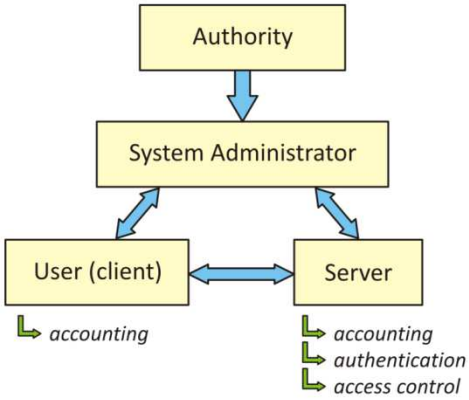

Figure 1: *Top level system diagram*

The resistor R_{SPU} is used for charging C_{ST} and it has many times lower resistance in comparison to R_{DPU}. On the slave side is a switch which controls connecting C_{ST} to 1W-line during charging. The line for the control switch uses negative logic (logic HIGH – switch ON and vice versa) and it is internally connected to a pull-down resistor RPD which causes the capacitor to be charged immediately after the voltage is plugged to 1W-line. The slave has to be able to pull down the 1W-line, this is achieved by nMOS transistor controlled by LOW control line. On the 1W-line is a diode that protects GPIO pin and its internally connected diode prevents over-voltage. This diode protects the microcontroller in the event the voltage on 1W-line pin is of a higher voltage than on Vcc.

The proper communication is conditioned by the detecting slave and charging its C_{ST} in the correct manner. This process is described in the diagram in the figure 3. The first step is launched in the master. The 1W-line is pulled up for a short time (approx. several microseconds). The voltage is measured immediately after that. In the event that the measured value is equal or very close to Vcc, the 1W-line is disconnected. Otherwise when the voltage is below *VDROP*, it means the interface is in short circuit state (between *1W-line* and *GND* is very low resistance). In both stated cases resistor R_{DPU} is disconnected via *DPU control line and the MASTER MCU* waits for 100 ms and previous steps are repeated. If measured voltage on 1W-line is in the range between higher than V_{DROP} and lower than V_{CC} than it means the slave has been detected. In this case R_{SPU} is connected to 1W-line and the possible current for charging C_{ST} is increased (due to $R_{SPU} << R_{DPU}$). In this state the master stands for 150 ms – time needed for full charging of 220 µF electrolytic capacitor. After that the master disconnects both pull-up resistors.

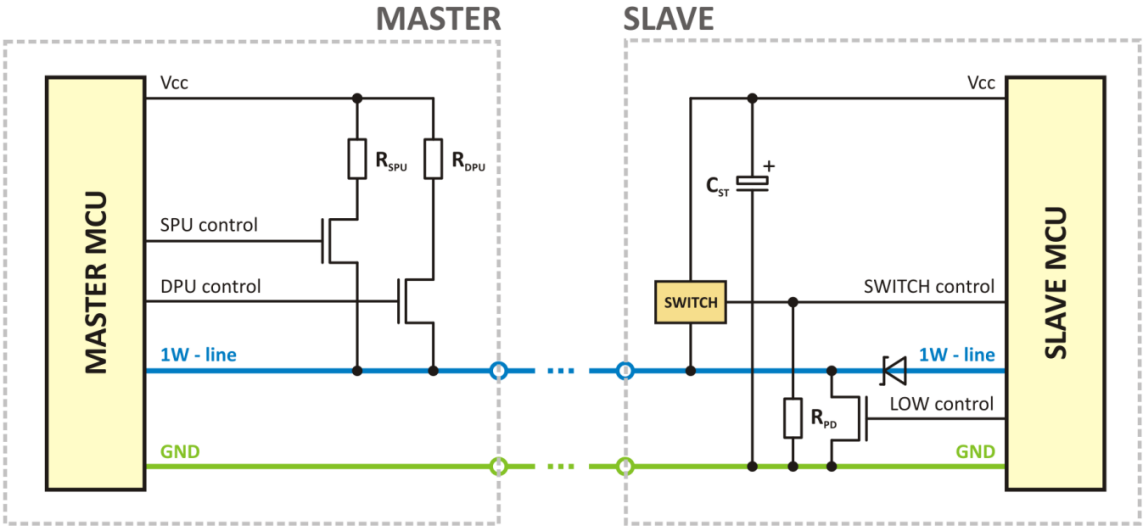

Figure 2: *Fundamental schematic of 1-Wire communication*

The slave side microcontroller starts to periodically measure voltage on the *1W-line* immediately after voltage on V_{CC} reaches 2 V (minimal supply voltage for microcontroller). After the slave detects a decrease of voltage on *1W-line* it means both pull-up resistors (R_{DPU} and R_{SPU}) on the master side have been disconnected. The slave reacts by switching off C_{ST} from the 1W-line.

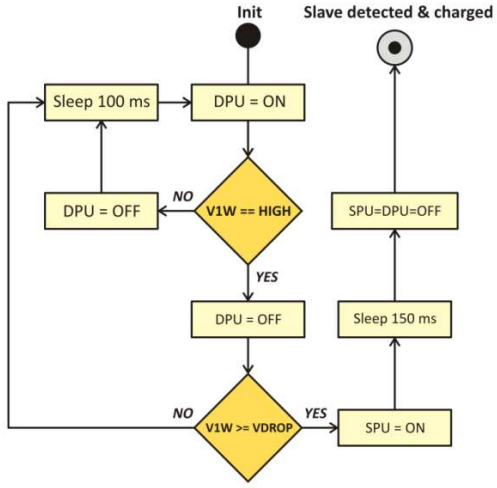

Figure 3: *Logical diagram of communication*

The timing diagram of the basic data exchange is in the figure 4. The process starts by detecting the slave and charging CST as previously described. The communication speed of 1-Wire interface is set to 6,4 kbps. After charging the master waits for 5 ms and gives the slave possibility to detect a decrease of voltage on 1W-line. This period is followed by resetting the 1-Wire interface and sending challenge from slave to master (from server to slave). The master processes challenge and sends back specific data to the slave (data are dependent on the task requested by the master). The slave performs authentication and authorization and then executes or not the secured service. During the time period dedicated for execution secured service RSPU is connected for recharging CST and for boosting power on the slave side. This final period takes 50 ms, but it could be changed for specific applications.

3. Authentication scheme

Any user in the system has an assigned unique identifier. The identifier is a number with the length of 4 bytes. All identifiers of users that can get access to secured service are stored in the list in the server memory. The current version of server firmware is able to store 100 identifiers. All entities of the system share the same encryption key K, in our case

it is 128 bits. The authentication process starts by sending challenge CH from server to master. The challenge is a 4 byte random number. This number is sent in a plain text form to the slave. The client creates a data packet which contains its identifier ID, challenge CH, task T and a random number RN with the length of 9 bytes. This packet is encrypted by key KC and sent back to the server. The server decrypts this packet by KS that should be equal to client. If KS is equal to KC the CH has to be the same as CH'. In this case the server can trust the ID' trying to find it in the list. In case this step was successful (list contains ID'), the server executes a secured service based on task T'. The client is informed about the result in the response R.

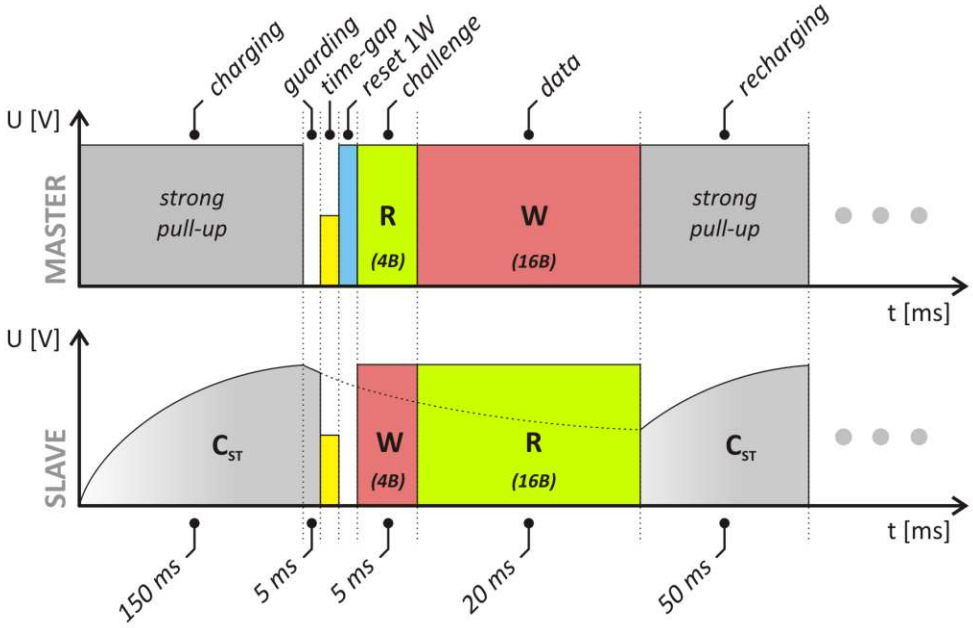

Figure 4: *Timing diagram of communication*

Table 1: *The authentication plan*

step	client (master)	data direction	server (slave)
1	CH	←	*CH*
2	C = *ENC*(*CH+ID+T+RN,K$_C$*)		
3	C	→	C
4			CH'+ID'+T'+RN' = DEC(C,K$_S$)
5			*R* = process(CH'+ID'+T')
6	*R*	←	*R*

4. Ferroelectric RAM

Flash memory technologies have several limitations for low power embedded environments. In the event writing to a nonvolatile memory is desired during application runtime flash memory could cause certain restriction [5], [7]. Mainly two factors restrict arbitrary use of flash memory. The first is endurance, where the number of write/erase cycles is finite. Nowadays flash memories are capable of 10000 cycles [6]. For realistic low power applications the power consumption during the write process could mean difficulties. The power consumption during writing could be 260 µA / MHz. Further the writing process requires certain voltage level that could be higher than available (discharged battery).

Possible solutions for above listed disadvantages could be seen in ferroelectric RAM technology. This memory type had been adopted for microcontrollers in 2012. The main differences between flash and FRAM are stated in table 2.

5. Rijandael Encryption

The AES 128 will be shortly brought back in this section. The comparison of software encryption and hardware accelerated encryption will be shown as well. The Rijandael algorithm is a symmetric block cypher specified in FIPS 197 (Federal Information Processing Standards Publication) [10], [11], generally well known as AES (Advanced Encryption System).The AES can encrypt or decrypt data blocks of 128 bits. Encryption converts data to an unintelligible form (cipher text) and decryption converts data to its original form.

Table 2: *Differences between Flash and FRAM technology*

	FRAM	Flash
Write speed (13 KB)	10 ms	1 s
Average active power [µA/MHz]	82	260
Write endurance [cycles]	10^{15}	10^{4}
Dynamic bit-wise programmable	yes	no
Unified memory (data/code)	yes	no

The algorithm works basically in four steps for each round. At first each byte of state matrix is replaced according to a lookup table. Next the last three rows are shifted cyclically for a certain number of steps. The third step is a mixing column. The method accepts four bytes as inputs and produces four bytes as output. Each input affects all output bytes. The last method adds a round key. The round key is derived from the main key and from the last round state. The AES128 10 rounds for encrypting one block cipher (16 bytes).

Below, two possible AES128 solutions for MSP430 microcontrollers will be compared. The comparison has been done on a CC430F5135 microcontroller [8]. This chip has an AES128 accelerating module. The first way is to use this module for encryption data. Unfortunately this method is not available on all MSP430 chips, due to the lack of AES128 accelerating modules. In this event we can use a software solution.

In this comparison, three parameters will be evaluated. In some applications time consumption could be very important (the time required for encryption or decryption of a certain volume of data). This parameter is closely related to power consumption (especially for battery powered applications this parameter is the most critical). The last observed parameter is the amount of memory occupied by encryption related functions in the flash memory space on the chip.

Table 3: *Comparison of time consumption*

operation	time consumption [ms]	MCLK cycles count [-]
encryption HWA	228	167
encryption SW	7960	6600
decryption HWA	256	214
decryption SW	10840	8400

note: data volume 10 kB, clock frequency 1MHz

Table 3 shows the time consumed for encrypting and decrypting 10 kB of data using our two ways. Hardware accelerated encryption is almost 35 times faster than software encryption. A similar situation could be observed in the decryption situation; its difference is a little bit higher there (42 times). In the last column there are counts of MCLK cycles announced in documentation for both methods.

All MSP430 chips are designed with a focus on low power consumption and they should be especially used in battery powered applications, same as in our system. Therefore each module and software part has to put an emphasis in reducing its power need as much as possible. Although the MSP430 architecture enables several low power modes, both of our methods for AES128 require an active mode. The microcontroller's core is powered up during this mode. The frequency source for the core could be used for the other peripheral modules on the chip. This is required for both our AES128 solutions. On tested CC430F5135 the power consumption is approximately 160 µA / MHz. The measured dependency of power consumption on clock speed is in figure 5. We can see we are not able to change power efficiency by changing clock speed. Therefore the frequency choice should be decided in respect to the other application requirements.

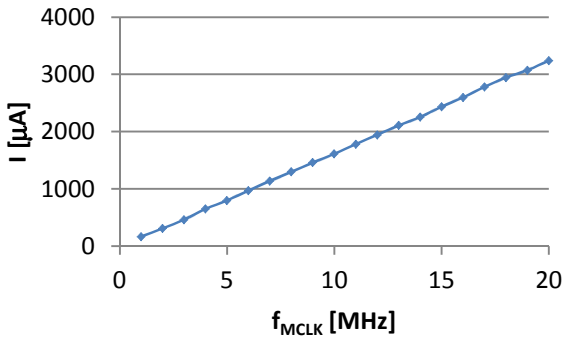

Figure 5: *Power consumption vs. clock frequency*

The last observed parameter is flash memory usage. In the event of the software method the requirements are far too ambitious. In some applications this factor could mean limitation, for example in the event flash memory is used for storing application data, not only code. You can see the comparison in the table 4.

The advantages of using a hardware accelerator module are visible in the short comparison stated above. Unfortunately not all MSP430 chips enable this module and there is no other way than a software solution. This could be satisfactory for an application where encryption resp. decryption is used just for authentication or the volume of data for encryption is limited. The other advantage of the hardware accelerator is that although the microcontroller has to be in active mode, the encryption could run in the background (just sourced from MCLK) and when it is done it will fire interruption or indicate interruption flag. In this case the microcontroller can do tasks concurrently (encryption in the background and the main task on MCU core).

Table 4: *Memory usage*

	size in program memory [B]	
	HW	SW
encryption method	42	1184
decryption method	42	1438
key expansion	204	550
sum	**288**	**3172**

6. Future Work

We plan to replace the current microcontroller for a newer type in the near future. The reason is the lack of AES acceleration module on the current type. The advantages of the hardware acceleration module are listed in this paper. As a result, we mainly expect a reduction in time wasted during software encryption and therefore a reduction of required power during authentication.

7. Conclusion

In this paper we have proposed an access management system for parasitically powered electronic devices. The authentication process uses symmetrical encryption algorithm AES 128. The proposed system is client – server oriented. The server is powered via 1-Wire communication interface to form the client. Both sides are able to log access events. The realized system is based on MCUs family MSP430FR that are FRAM enabled. The features and advantages of the FRAM nonvolatile memory are listed in this paper. In future work we would like to replace the current MCUSs with a newer version which has an AES encryption acceleration module.

8. Acknowledgment

Research described in this paper was financed by the National Sustainability Program under grant LO1401. For the research, infrastructure of the SIX Center was used.

References

[1] K. Takahashi, Identity Management: Concepts, Technologies, and Systems (Information Security & Privacy), Artech House Publishers, 2011, p. 196.

[2] R. He, M. Yuan, J. Hu, H. Zhang, Z. Kan and J. Ma, "A Novel Service-Oriented AAA Architecture," *Personal, Indoor and Mobile Radio Communications,* no. vol.3, pp. 2833-2837, 2003.

[3] V. Shatuka, P. Banga and B. Carroll, AAA Identity Management Security, Cisco Press, 2010, p. 480.

[4] R. E. Cnith, Authentication: From Passwords to Public Keys, Addison-Wesley Professional, 2001.

[5] E. Osmanoglu, Identity and Access Management: Business Performance Through Connected Intelligence, Syngress, 2013.

[6] MAXIM, "MAXIM 1-Wire Presentation," [Online]. Available: http://www.maximintegrated.com/en/products/1-wire/flash/overview/. [Accessed 12 Februrary 2015].

[7] H.-L. Li, Y. Chia-Lin and H.-W. Tseng, "Energy-Aware Flash Memory Management in Virtual Memory System," *IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS,* pp. 952-964, 18. 7. 2008.

[8] D. Takashima and I. Kunishima, "High-density chain ferroelectric random access memory (chain FRAM)," *The IEEE Journal of Solid-State Circuits,* pp. 787-192, May 1998.

[9] R. Bailey, G. Fox, J. Eliason, M. Depner, D. Kim, E. Jabillo, J. Walbert, S. Summerfelt, K. Udayakumar, J. Rodriquez, K. Remack, K. Boku and J. Gertas, "FRAM Memory Technology - Advantages for Low Power, Fast Write, High Endurance Applications," *Proceedings of the 2005 International Conference on Computer Design,* p. 485, 2005.

[10] "ADVANCED ENCRYPTION STANDARD (AES)," Federal Information Processing Standards Publication 197, 2001.

[11] T. P. Mussolini, "Integration of IPs into the M8051 microcontroller," *Programmable Logic (SPL), 2012 VIII Southern Conference,* pp. 1-6, 2012.

[12] Texas Instruments, "Datasheet CC430F5137," [Online]. Available: http://www.ti.com/lit/ds/symlink/cc430f5135.pdf. [Accessed 5. 2. 2015].