

Technical status Zifra card

Torbjörn

May 30, 2018

Contents

1	todo	3
2	Brief description	3
3	System overview	3
3.1	Key generation	3
3.2	Encryption	5
3.3	Decryption	6
4	Hardware	6
5	sd slave progress	6
6	chacha	7
7	Helpful data	7
7.1	CSD list	7
8	Whishlist	10
8.1	Hidden files	11
8.2	See pictures when camera is still on	11
8.3	Fake pictures	11

Abstract

This document will summarize have bin done and what is needed to do to get to the final product.

1 todo

Here i will list stuff i come up whit so that i can write about them later.

- crypto scheem whit curv 25519
- Alex algorithm
- hiding files in fat
- fat format

2 Brief description

We will here shortly describe what the Zifra card is meant to be. It is an memory card that is supost to protect data by encrypting the data that is writen to it. This is done whit the help of multiple key(s) and cipher. i

3 System overview

We will here describe the way we want the system to work. Fore the case of explenation we will use an camera as example case. We will first talk about the key generation secondly encryption and finally about decryption step.

3.1 Key generation

There is three type of keys used in the crypto scheme. The first two keys are pre-generated as an key pair on an safe device one private key (Pri_key) and one Public key (Pub_key). The pri_key is saved fore later use and the pub_key is transfered and saved on the sd-card see Figure 1. Then the user puts the sd card in an camera. When the camera boots up an random number is generated this is the Random generated session Key (Rand_key) this number is then encrypted whit the Pub_key and saved on the memory.

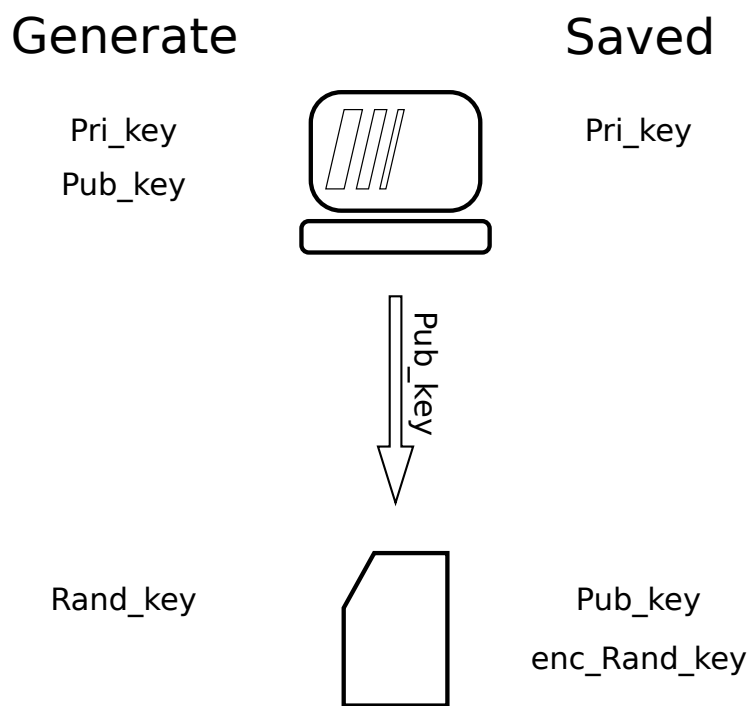


Figure 1: This illustration shows where the keys are generated and where they are stored

3.2 Encryption

The Rand_key¹ is then used as the key input in the chacha cipher to encrypt the data before it is saved to the memory se Figure 2. When the system is powered of the Rand_key that is only stored in the FPGA disappeared due to the nature of an FPGA. The generation and encryption of the Rand_key is repeated fore each session. So there will be multiple enc_Rand_key:s stored on the memory one fore each session.

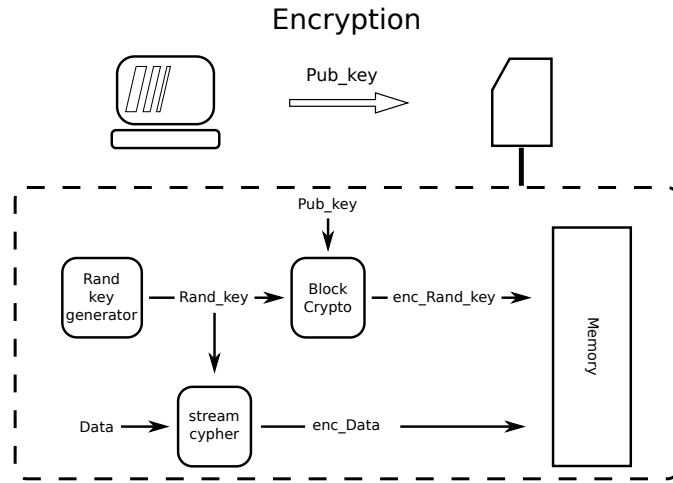


Figure 2: Basic data flow in encryption step

¹This is the non encrypted random key.

3.3 Decryption

When decrypting the data one extracts the encrypted data and the Encrypted Random generated session Key (enc_Rand_key) on to the safe device were the Pri_key is saved. Then we use the Pri_key to decrypt the enc_Rand_key:s and then we can use the Rand_key:s to decrypt the data se Figure 3.

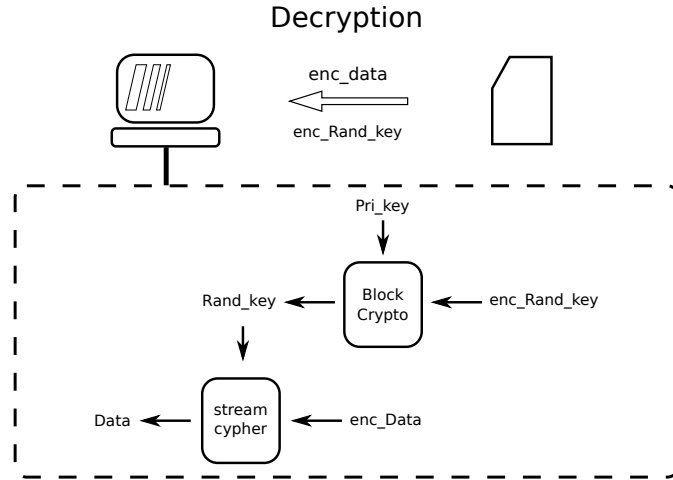


Figure 3: basic data flow in decryption step

4 Hardware

The hardware was intended to be an Printed Circuit Board (PCB) whit an Field Programmable Gate Array (FPGA) whit an memory and some suport component. The FPGA that we use is the Artix7 t35.

5 sd slave progress

The sd slave comes form the Google vault project [2]. This part of the project dose not work yet.

We have extracted the blocks that we think are needed fore the project. There test bench to test and send commands to the design and see what re-sponses the device sends back. So fare we can read and write from the first sector in the memory but not to any other sector.

6 chacha

The stream cipher that was decided to use for the data encryption is called chacha [1] developed by Joachim Strömbergson [3]. We have made an axi ip block out of it one stream port for the data and one address port for the setting example iv and key see Figure 4. There is a test bench that shows how the axi block is intended to be used.

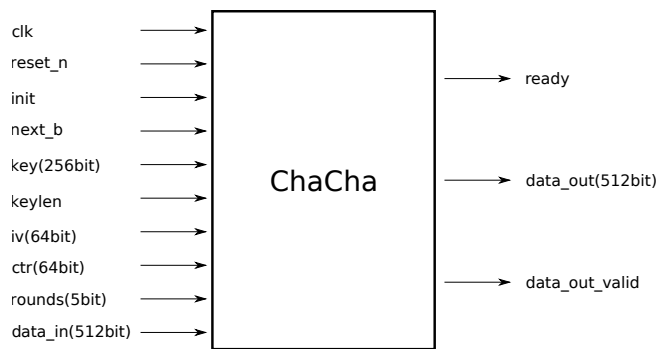


Figure 4: Port configuration chacha cipher

7 Helpful data

This section will list information that has been collected during the project that might help further development.

7.1 CSD list

During the debugging we listed the CSD values that were used in the `gv_sd_slave` we also made a list of the other CSD values.

add the CSD tables here

This is the list from part1 Version (1.0)		
Name	Description	CSD-slice
CSD structure	CSD structure	[127:126]
-	reserved	[125:120]
TAAC	data read access-time-1	[119:112]
NSAC	data read access-time-2 in CLK cycles (NSAC * 100)	[111:104]
TRAN_SPEED	max. read data block length	[103:96]
CCC	card command classes	[95:84]
READ_BL_LEN	max. read data block length	[83:80]
READ_BL_PARTIAL	partial blocks for read allowed	[79:79]
WRITE_BLK_MISALIGN	write block misalignment	[78:78]
READ_BLK_MISALIGN	read block misalignment	[77:77]
DSR_IMP	DSR implemented	[76:76]
-	reserved	[75:74]
C_SIZE	device size	[73:62]
VDD_R_CURR_MIN	max.read current @ VDD min	[61:59]
VDD_R_CURR_MAX	max.read current @ VDD max	[58:56]
VDD_W_CURR_MIN	max.write current @ VDD min	[55:53]
VDD_W_CURR_MAX	max.write current @ VDD max	[52:50]
C_SIZE_MULT	device size multiplier	[49:47]
ERASE_BLK_EN	erase single block enable	[46:46]
SECTOR_SIZE	erase sector size	[45:39]
WP_GRP_SIZE	write protect group size	[38:32]
WP_GRP_ENABLE	write protect proup enable	[31:31]
-	reserved	[30:29]
R2W_FACTOR	write speed factor	[28:26]
WRITE_BL_LEN	max. write data block length	[25:22]
WRITE_BL_PARTIAL	partial blocks for write allowed	[21:21]
-	reserved	[20:16]
FILE_FORMAT_GRP	file format group	[15:15]
COPY	copy flag	[14:14]
PERM_WRITE_PROTECT	premanet write protection	[13:13]
TMP_WRITE_PROTECT	temporary write protection	[12:12]
FILE_FORMAT_GRP	file format	[11:10]
-	reserved	[9:8]
CRC	crc	[7:1]
-	not used always '1'	[0:0]

This is the list from part1 Version (2.0)		
Name	Description	CSD-slice
CSD structure	CSD structure	[127:126]
-	reserved	[125:120]
TAAC	data read access-time-1	[119:112]
NSAC	data read access-time-2 in CLK cycles (NSAC * 100)	[111:104]
TRAN_SPEED	max. read data block length	[103:96]
CCC	card command classes	[95:84]
READ_BL_LEN	max. read data block length	[83:80]
READ_BL_PARTIAL	partial blocks for read allowed	[79:79]
WRITE_BLK_MISALIGN	write block misalignment	[78:78]
READ_BLK_MISALIGN	read block misalignment	[77:77]
DSR_IMP	DSR implemented	[76:76]
-	reserved	[75:70]
C_SIZE	device size	[69:48]
	reserved	[47:47]
ERASE_BLK_EN	erase single block enable	[46:46]
SECTOR_SIZE	erase sector size	[45:39]
WP_GRP_SIZE	write protect group size	[38:32]
WP_GRP_ENABLE	write protect group enable	[31:31]
-	reserved	[30:29]
R2W_FACTOR	write speed factor	[28:26]
WRITE_BL_LEN	max. write data block length	[25:22]
WRITE_BL_PARTIAL	partial blocks for write allowed	[21:21]
-	reserved	[20:16]
FILE_FORMAT_GRP	file format group	[15:15]
COPY	copy flag	[14:14]
PERM_WRITE_PROTECT	permanent write protection	[13:13]
TMP_WRITE_PROTECT	temporary write protection	[12:12]
FILE_FORMAT_GRP	file format	[11:10]
-	reserved	[9:8]
CRC	crc	[7:1]
-	not used always '1'	[0:0]

Settings from GV sd_parms.vh		Comment
Name	Value	
CSD_CSD_STRUCTURE	b01	
CSD_TAAC	h0E	
CSD_NSAC	h00	
CSD_TRAN_SPEED_25	h32	
CSD_TRAN_SPEED_50	h5A	
CSD_CCC	b01_1_110110101	
CSD_READ_BL_LEN	9	512 byte Blocks
CSD_READ_BL_PARTIAL	h0	Partial block reads not i
CSD_WRITE_BLK_MISALIGN	h0	Write misalingned block
CSD_READ_BLK_MISALIGN	h0	Read misaligned block n
CSD_DSR_IMPL	h0	DSR not implemented
CSD_C_SIZE	8	4.5 MBYTE
SD_TOTAL_BLOCKS	(CSD_C_SIZE+1) * 1024	9216
CSD_ERASE_BLK_EN	h1	
CSD_SECTOR_SIZE	h7F	
CSD_WP_GRP_SIZE	h0	
CSD_WP_GRP_ENABLE	h0	
CSD_R2W_FACTOR	b010	
CSD_WRITE_BL_LEN	h9	
CSD_WRITE_BL_PARTIAL	h0	
CSD_FILE_FORMAT_GRP	h0	
CSD_COPY	h0	
CSD_PERM_WRITE_PROTECT	h0	
CSD_TMP_WRITE_PROTECT	h0	
CSD_FILE_FORMAT	b00	
Settings from GV sd_const.vh		
Name	Value	
CMD9_SEND_CSD	d9	
CMD27_PROGRAM_CSD	d27	
STAT_CSD_OVERWRITE	d16	

add cid by hand

8 Whishlist

Here we will list some features that we wold like to have in the future. Thees are extras but good to have.

8.1 Hidden files

We would like to have the possibility to hide the files so that every time you start up the memory card it looks empty. This would be done by making an temporary FAT table that is used during the session then hidden between the MBS and the real FAT table.

8.2 See pictures when camera is still on

We would like to be able to look at the pictures during the session. This would mean that the cipher needs to work in both directions during the session.

8.3 Fake pictures

We would also like to be able to prep the memory card with fake data in the camera case that would be preloaded with pictures. These would show up if someone would inspect the card. This would also be helped by the see pictures while session is active if the system is tested.

Glossary

cipher Is an mathematical algorithm that takes two inputs data and an key and outputs an new data string that can not be read whit out an corresponding key. 3

gv_sd_slave This is the sd_slave from the google vault project. 7

key(s) This refers to an sires of bits used as an input to the cipher. 3

MBS Master boot record is an boot sector in the beginning of an memory that tells the system where to start. 10

Acronyms

enc_Rand_key Encrypted Random generated session Key. 6

FPGA Field Programmable Gate Array. 6

PCB Printed Circuit Board. 6

Pri_key private key. 3

Pub_key Public key. 3

Rand_key Random generated session Key. 3

References

- [1] ChaCha. <https://git.cryptech.is/core/cipher/chacha.git>. Accessed: 2018-05-22.
- [2] GoogleVault. <https://github.com/ProjectVault/orp>. Accessed: 2018-05-22.
- [3] joachim strömbergson. Chacha (cypher).