

Some notes about EWS3 and EWS4:

Overall:

- no reason to explain situations where CAS is plain (not crypted - SOPT not activated), because almost anybody can solve all common problems with them. Lot of key makers work with plain CAS dump, ISN and SK both are "in plain text".
- not a great reason to explain situations if it is possible to get real SK from ECU without any pain - if you have real values, you can easily get all 3 bytes of cryptokey and decrypt dump.
- even if CAS is crypted but valid key exists, it is possible to read real Page3 from key using cryptoLO and cryptoHI from CAS dump. This real Page3 you can use to decrypt entire CAS dump and get real ISN and SK.

Notes:

- **ISN** = Individual Serial Number. 1 ½ bytes (12 bits) for EWS2, 2 bytes for EWS3 ECU. Complete ISN is 4 bytes long and used for 6HP EGS (even on first F-series cars).
- **SK** = Security Key, 16 bytes long. Used for EWS4 and up.
- **Cryptokey** - 3 bytes, used for CAS decryption. Not official term...
- EWS2, **EWS3**, **EWS4**, EWS5 - we are talking about protocols, not physical hardware.

EWS3:

necessary tools – **CAScontrol**, **Htag**:

If ECU works using ISN (EWS3) and there is no 6HP EGS on car, task is simple:

- at first calculate real ISN (2 bytes from 4, ones necessary for ECU). You can do that via rolling codes using "ISN restore from rolling codes" utility from CAScontrol set. Remaining 2 bytes can be any, usually they are 0000, complete ISN is used only for EGS;
- load CAS dump into CAScontrol CAS dump tool, decrypt it using real ISN. All possible cryptokey values are good enough, use any;
- build key according to key data, start a car.

If ECU works using EWS3 protocol and there is a 6HP EGS with activated EWS mounted on car:

- you can get 2 bytes of complete ISN as written above;
- you can calculate missing 2 bytes from EGS RAM using **CAScontrol** tool;
- now you can decrypt CAS dump and make a key. As in previous case - any cryptokey is good enough.



You can use **HTag** emulator to test results, just "build" any already used and not blocked key.

EWS4:

necessary tools: **CAScontrol**, **Htag**, **EWS4 drive authorization unit**, **EWS4 8HP authorization unit** (EWS4 emulators), **CAN logger** or **MBcan** tool for configuration purposes.

Some common situations from practice:

- all keys lost, CAS is crypted, ECU works on EWS4, not readable. Must make key or get real SK for ECU,
- you got used EGS, you have CAS without keys, CAS is crypted.

How to proceed:

- usually there is untouched ISN backup in plain inside CAS dump - in case of CAS3+ on XDP it is located at address 0x010. If CAS dump is not damaged, this value remains untouched. If somebody switched CAS back to EWS3 from EWS4 and tried to make alignment, it is a great problem...
- find all cryptokeys according to real ISN, get all possible SK. There will be 16 or 64 possible values (depends on crypto type);

- now must test all keys using EWS4 emulator. In worst case it will take about 1 hour to test them all:
 1. write possible SK into EWS4 emulator, switch OFF/ON
 2. observe EWS status with Tool32:
 - **Status 0:** no authorization request / response - verify connections and jumper settings!
 - **Status 1:** authorized, SK is valid
 - **Status 2:** invalid SK, no authorization
 - **Status 3:** SK not defined - in case of EGS if key agreement take place and SK is most likely valid. After next power-on must go to status 1
- in case of ECU you can test all possible cryptokey values using **HTag** too. Not so fast method, complete car is necessary. But it works!
- now you can do what you wish - SK is valid, you can decrypt dump, make key, transfer SK to another CAS etc.

Custom built cable for 8HP EGS:

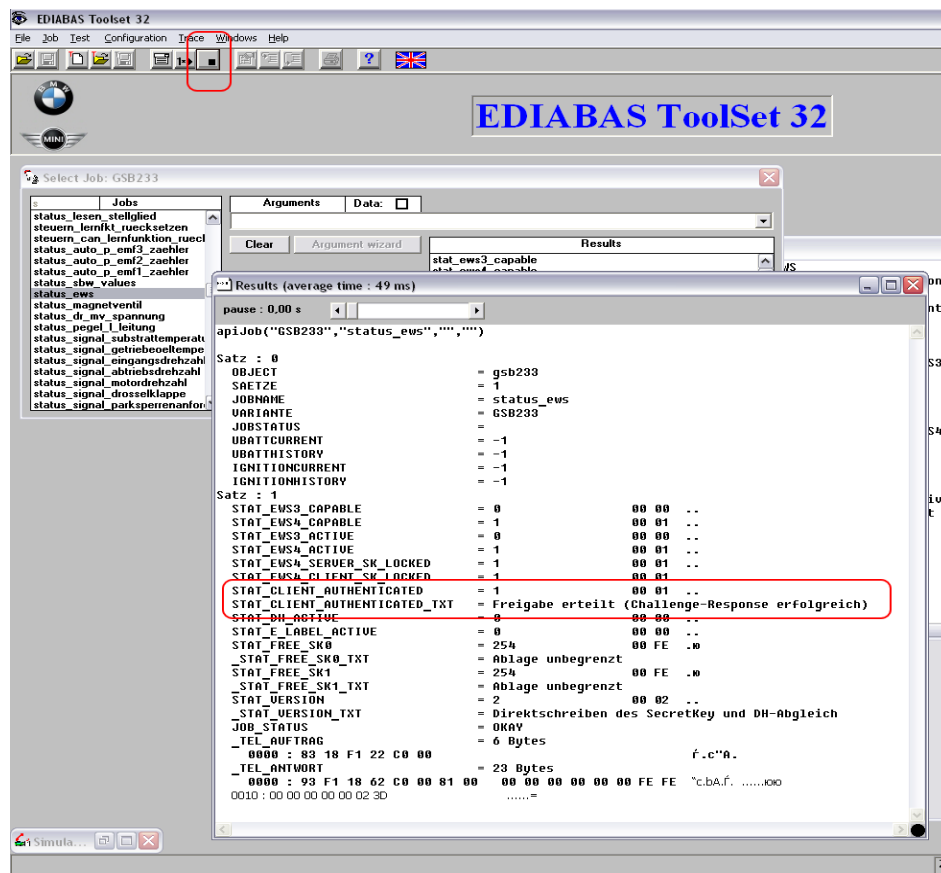
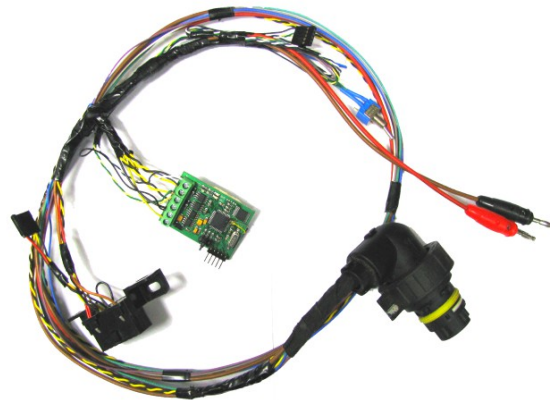
From EGS connector:

- **CAN-Hi** and **CAN-Lo** directly to EWS4 emulator CAN2 port,
- **+12v**,
- **15WUP** via rocker switch to +12,
- **GND**.

Other wires are not mandatory.

OBD connector for **DCAN** adapter and / or **MBcan** for SK write - connected to EWS4 8HP emulator CAN1 port;

It is possible to connect **DCAN** adapter and **MBcan** both to OBD socket using Y-cable.



Some things to know and understand:

- authorization chain: KEY ↔ CAS ↔ ECU ↔ EGS. According to ideology, only authorized ECU can authorize EGS.
- only authorized CAS can authorize ECU;
- SK used for CAS and ECU relations is SK_SERVER for CAS, but SK_CLIENT for ECU;
- in Tool32 screen (above) STAT_CLIENT_AUTHENTICATED mean EGS is a CLIENT here.

SK in emulators:

- EWS4 emulator emulates CAS for ECU, SK_SERVER here.
- EWS4 8HP emulator emulates ECU for EGS, SK_CLIENT is for relations with CAS, SK_SERVER is for relations with EGS.
- SK_SERVER is a result of key agreement procedure between ECU and EGS.

EWS4 configurator for MBcan hardware:

- read / write SK_CLIENT,
- read / write SK_SERVER

It is possible to use configurator with both kinds of EWS4 emulators (EWS4 or EWS4 8HP).

Must note that into EWS4 emulator for ECU (which emulates CAS) you can store only SK_SERVER.

Example:

what we had:

- EGS 8HP and
- CAS without any key, crypted and updated to latest I-level

what we did:

- it was possible to recover ISN from CAS dump;
- we made a list of all 64 possible SK using CAScontrol utility (it is a worst case, old crypto gives only 16 possible results);
- prepare hardware: EGS itself, cable (with EWS4 8HP emulator), **DCAN** and **MBcan** via Y-cable, power supply, 2x PC - one for Tool32 in continuous monitoring mode, other one for **MBcan**. Really one PC is enough, but to use 2 is much better.

Now at each step (process every possible SK) :

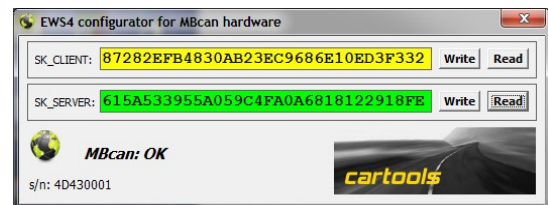
- **emulator:** jumper to configuration mode;
- write SK;
- **emulator:** jumper to key agreement mode;
- power OFF;
- power ON - must restart EGS. It is possible do that via Tool32 too (job: [egs_reset](#));
- observe what happens with authorization.

Note that key agreement is possible only at EGS startup and is always initiated by EGS itself. If nothing happens, must clear EGS errors or make a larger delay between power OFF and ON.

As you can see, our correct one was located somewhere in the middle of entire list.

For testing one SK about 1 minute is necessary. So in worst case overall process can take about one hour.

After key agreement procedure SK_SERVER is generated (common key) and stored into both EGS and EWS4 8HP emulator. You can read it out via configurator now.



©

```
1BB4B267D4AC37BF700A1A7D924F6FAE *
14BBBD68DBA338B07F0515729D4060A1 *
11BEB86DDEA63DB57A001077984565A4 *
0FA0A673C0B823AB641E0E69865B7BBA *
0AA5A376C5BD26AE611B0B6C835E7EBF *
05AAAC79CAB229A16E1404638C5171B0 *
00AFA97CCFB72CA46B110166895474B5 *
3C939540F38B1098572D3D5AB5684889 *
39969045F68E159D5228385FB06D4D8C *
36999F4AF9811A925D273750BF624283 *
339C9A4FFC841F9758223255BA674786 *
2D828451E29A0189463C2C4BA4795998 *
28878154E79F048C4339294EA17C5C9D *
27888E5BE8900B834C362641AE735392 *
228D8B5EED950E8649332344AB765697 *
5FF0F62390E873FB344E5E39D60B2BEA *
5AF5F32695ED76FE314B5B3CD30E2EEF *
55FAFC299AE279F13E445433DC0121E0 *
50FF9F2C9FE77CF43B415136D90424E5 *
4EE1E73281F962EA255F4F28C71A3AFB *
4BE4E23784FC67EF205A4A2DC21F3FE *
44EBED388BF368E02F554522CD1030F1 *
41EEE83D8EF66DE52A504027C81535F4 *
7DD2D401B2CA51D9166C7C1BF42909C8 *
78D7D104B7CF54DC1369791EF12C0CCD *
77D8DE0BB8C05BD31C667611FE2303C2 *
72DDDB0EBDC55ED619637314FB2606C7 *
6CC3C510A3DB40C8077D6D0AE5381D9 *
69C6C015A6DE45CD0278680F3D1DDC *
66C9CF1AA9D14AC2D0776700BF321D3 *
63CCCA1FACD44FC708726205EA3717D6 *
9C3335E0532BB038F78D9DFA15C8E829 *
993630E5562EB53DF28898FF10CDED2C *
96393FEA5921BA32FD8797F01FC2E223 *
933C3AEF5C24BF37F88292F51AC7E726 *
8D2224F1423AA129E69C8EB04D9F938 *
882721F4473FA42CE39989E01DCFC3D *
87282EFB4830AB23EC9686E10ED3F332 !!!!!!!!! OK
822D2BFE4D35AE26E99383E40BD6F637 *
BE1117C27109921AD5AFBFD837EACA0B *
BB1412C7740C971FD0AABADD32EFCF0E *
B41B1DC87B039810DFA5B5D23DE0C001 *
B11E18CD7E069D15DAA0B0D738E5C504 *
AF0006D36018830BC4BEAEC926FBD81A *
AA0503D6651D860EC1BBABCC23FEDB1F *
A50A0CD96A128901CEB4A4C32CF1D110 *
A00F09DC6F178C04CBB1A1C629F4D415 *
DD7274A1126AF179B6CCDCBB5489A968 *
D87771A4176FF47CB3C9D9BE518CAC6D *
D7787EAB1860FB73BCC6D6B15E83A362 *
D27D7BAE1D65FE76B9C3D3B45B86A667 *
CC6365B0037BE068A7DDCDAA4598B879 *
C96660B5067EE56DA2D8C8AF409DDB7C *
C6696FBA0971EA62ADD7C7A04F92B273 *
C36C6ABF0C74EF67A8D2C2A5A97B776 *
FF5056833048D35B94EEFF9976AB8B4A *
FA555386354DD65E91EBFF9C73AE8E4F *
F55A5C893A42D9519EE4F4937CA18140 *
F05F598C3F47DC549BE1F19679A48445 *
EE4147922159C24A85FFEF8867BA9A5B *
EB444297245CC74F80FAEA8D62BF9F5E *
E44B4D982B53C8408FF5E5826DB09051 *
E14E489D2E56CD458AF0E08768B59554 *
1EB1B762D1A932BA750F1F78974A6AAB
```

RobinDAB '2016...'2026 for www.cartools.lv