

IPENCRYPTER.COM

Encrypt HDL Intellectual Property (IP)

ipecrypt

compliant with
IEEE Std 1735™-2023 standard

IP Encrypter

Table of Contents

Table of Contents.....	1
Encrypt IP	2
Download.....	2
<i>ipecrypt</i> command syntax for encryption.....	2
1. Process of creating encrypted IP using annotated IP	3
1.1 Create IP in plain text.....	3
1.2 Add protect directives	3
1.3 Run <i>ipecrypt</i> to generate encrypted IP	5
2. Alternative ways to add protect directives.....	6
2.1 YAML.....	6
2.2 JSON	7
2.3 Plain text	9

Encrypt IP

ipecrypt is an application to encrypt the HDL IP. An IP author provides the information to protect IP through `protect` directives. The author can target a set of tools for encrypted IP. The author requires public key information for each tool.

Download

The *ipecrypt* is available for download at:

<https://ipencypt.com/downloads/hdl-ip-encrypter-tools>.

For license proxy and decryption products conforming to the “IEEE Std 1735™-2023” visit:

<https://ipencypt.com>

***ipecrypt* command syntax for encryption**

The basic command for encryption is:

```
ipecrypt --infile <input> --outfile <output>
```

Complete list of arguments:

-h [--help]	produce help message
-I [--infile] arg	input file to encrypt
-O [--outfile] arg	output file encrypted
-F [--force]	force overwrite
-D [--directive] arg	directives file; directives from the file will be used for encryption. Use YAML (.yaml), JSON (.json) or text (.txt) file formats
-L [--language] arg	target language "verilog" or "vhdl"; it is required if directives information is provided through YAML or JSON directive file. It is optional if it is through text file.

1. Process of creating encrypted IP using annotated IP

1. Create IP in plain text
2. Add `protect` directives around the code to encrypt
3. Run *ipecrypt* application to generate encrypted IP

1.1 Create IP in plain text

Here is an example of a simple counter implemented in Verilog (`counter_p.v`):

```
*****  
// Eight-bit Counter  
// ipencypt.com  
*****  
module counter (out,clk,enable,reset);  
    output [7:0] out;  
    input      clk, enable, reset;  
    reg [7:0]   out;  
  
    always @ ( posedge clk )  
        begin  
            if ( reset )  
                out <= 8'b0 ;  
            else if ( enable )  
                out <= out + 1;  
        end  
endmodule // counter
```

1.2 Add protect directives

Add `protect` directives to the section of the IP to encrypt.

For Verilog use `'pragma protect` directive keyword and for VHDL use `'protect` directive keyword.

The counter with `protect` directives is (`counter_2e.v`) below:

```
*****  
// Eight-bit Counter  
// ipencypt.com  
*****  
'pragma protect version=3  
'pragma protect author="IP Encrypter"  
'pragma protect author_info="IPEncryter.com"  
'pragma protect data_method="aes256-cbc-hmac-sha512"  
  
'pragma protect begin_commonblock  
'pragma protect license_proxyenv="IPE_V3_PROXY"  
'pragma protect license_certificate  
MII FwzCCA6sCAQEwDQYJKoZIhv...  
VQQIDApDYWxpZm9ybmlhMREwDwYDVQQHDAhTYW4gSm9zZTEYMBYGA1UECgwPaXB1  
bmNyeXB0ZXIuY29tMRYwFAYDVQQLDA1JUCBQcm90ZWN0aW9uMRUwEwYDVQQDAXJ  
UCBFbmNyeXB0ZXIxIzAhBgkqhkiG9w0BCQEWFGLuZm9AaXB1bmNyeXB0ZXIuY29t  
MB4XDTIwMDQwNjA1MjAwM1oXDTMwMDQwNDA1MjAwM1owgaoxCzAJBgNVBAYTA1VT  
MRMwEQQDVQQIDApDYWxpZm9ybmlhMREwDwYDVQQHDAhTYW4gSm9zZTEYMBYGA1UE  
CgwPaXB1bmNyeXB0ZXIuY29tMRYwFAYDVQQLDA1JUCBQcm90ZWN0aW9uMRwwGgYD  
VQQDBNJuCBFbmNyeXB0ZXIgQ2xpZw50MSMwIQYJKoZIhv...  
ZW5jcnlw...  
iGzVDVr4F7u+TSqYD4o4G0Li2mKdpgO2FXufEnidSTdwVls8sY71lyJFO5114TYR  
ctKC...  
fYuNJiytu
```

```

U9SzxmTHH3So6gv/f3Aict9xGAL8O92vJtDTiim805Tx0max4isQvEXteCfcJpGe
tW7Z377Crp6sOKKY98170G/SI1QAsvK1ydgMJaPQeICSnBZJp1/y8G6K3oNJGRjn
W+F5ZtcPh2H2MMrIM0Yj6oaA2xbVaKyBzCdycnOAqkSGTJ5+ajsLGpDdn88BfEJ5
G9t7juI/Eop5azYrDLvoHipe9CVHjnwl+njwrn4PAqnPdpfv9GzPfVNh+7r74qR2
pEWlsw/8Sz090k+qsh3HO3RIUvDfRnzvmLr0fxp3Ljmdxe4rozdyGUbvs3jKEhyz
/t49GtOL7MxRC61JYJlmUHk2U+2oizPUXuvBMM3C4Elc5I+11cLOJTppY4mF9oz2
kh++G/QUnHFip2J4GqgPBoDpBFFnLch1r0JMZHehhSji+l/tpJcq3nri8+iZQbsO
FmY1SERu8SwT7+TETyG76H+sfp5RtB1JLO+wBtpY8e6jPzqy/MpJT4mak/kDovG
XDagdhSHO2v8VDufiC5WAtMiLKQHwYRppe0JtjDAGMAAEwDQYJKoZIhvcNAQEL
BQADggIBAAIxIpt1AMh/zPQc2EdZd8q37k14jPbXX1X3mQFTBMWgEmxmpg9ywhps
4drGyzZrhck1t7p2PhruOJRNgfZNFsDKBKY/YuuHyXOkgLCPFTWotDEi/coj391
1fWnN5oM/+fm3oWnUQ1mYDP/WoAkzp87tSHTxj/Z2pD0IBzTUnNa5lyfinCTEXXO
+UoRE66Bqe3JiQhRTbVAjvZ/r/m1MVkabUFaNH9g5VzuU4pu1D8JLdq6SBMXk0rE
Q3+hGPynQc4d5Ns1P6ko1ADRbjBoqMyTrjTOg4/Ysb06jJMrv1VI+USDS3poma8y
cMYK6hkss90i8gfmp8ZmFje69Li1J1U+ibPY0imoS0M2n2DJ5vHm6Hmr+Zw6TkP
ncp6Iz3YzKV8P/Pxp+ilakDrrDA57tuABKD16bkSoRW40ZSlnJ2SR30FaF/BvHcf
nCKo/aDDFan/5C3qSFpseDqI34TXZeQyAOIE9qGv1rBeQIrdPlnLSkGKmqj/vp9e
NAkePO9smchle4Eoh8kTFM5B3VtMt4cy60YaOHGUvIzAyORErzxuuuFf+yfbWWPa
RmB3q2h9HV4QPtkkAytVkuXfyCGeWltfojY2SBHQhzOdR/pBbxDlc2AxXuKnv9
JK7gv7TJyHa7dTFOYAWCfrwAlvetx5qGHikWaiUHXeMKK9ooR838
`pragma protect end_commonblock

`pragma protect begin_toolblock
`pragma protect key_keyowner = "ip encrypter"
`pragma protect key_keyname = "ip encrypter_key1"
`pragma protect key_method = "rsa"
`pragma protect key_public_key
MIICIjANBgkqhkiG9w0BAQEFAOCg8AMIIICCgKCAGEAzkwR73KUII/a6khZXDvz
7/cgzSMdkGq2xbNQQ/gvmJVFBuM+Nv1wjJcZmS7cJwLwA0LvV8EqWcy3Wqc3cceT
EFj1TZsMswrcGAaPrI51i0ewAOn1VPNfQT8fQJFkPdJpLB62Q8BaYvUP9KmAVCK6
DeCWGKYf0yBGjG411AeYxQ9VuVpIzPS06eZ0xH81EXR1s8loy60kzGG0WRbR29FO
YY2ti7HyWvAHsbURPZ6PhGM62P2X3rsO2obGQbE0d52X5BAy9G6oAAUY+FP1QD
oR05Xryfecdvan3OmclkxD+G2USeCJ6DI+fyqkQH1qyV7sISTgnEzE+wcW5phCAZ
k4ZC7zItnRJbVN1NHSSiiJq3pT3s8gf9ohMI9F72aWhM3DKCNX/D41qULjsYvg
I2C8pr6ILVdE3da4Ueq9Z/9aARWGyMfd0pQ7qkieV/YVn5rWAFTNLiuVyyI9oEZR
mRnKZF3N+SEdLnPXATTi004t9svn3zDZ7WHKx7GT5/IYe6KmUZ38CHOXMUTJKQqR
zXv+5kKDRvRD1EkTykkh12zGyUfizo50tUy2OXwo8ZfqEtGcob1GqpgOVQ0+us1/
z64cd612/RMkix2B9ETPudMeP8UNHBKz2zRnEsadUquVu+SUzfUPkTv1Svo4Hx
aSPWSriewi0V9UYKNQXuOvsCAwEAAQ==
`pragma protect end_toolblock

`pragma protect begin_toolblock
`pragma protect key_keyowner = "ip encrypter"
`pragma protect key_keyname = "ip encrypter_key2"
`pragma protect key_method = "rsa"
`pragma protect key_public_key
MIIBIjANBgkqhkiG9w0BAQEFAOCg8AMIIIBCgKCAQEA2xEbEbK+w5BWTBoOLQRB
qyK2hYWRC2z85eEncog6pyIisXeoIFCOKfVVCCBm6u+ebKJqmvN81hmkKICJv7pb
mO/LWJEQwyDsOqJYquTDMC+zqmqc7DuGiuz8/XmVpiziDtquPVDztXdnFyFZLMq
wEBna0zOMWTMZjb2LozU2jqy04vwuESAV4f1Nk1/96KZUp7pZF1X1jjpNY5UoHAt
20Ysn35zdebKdNp1SKAHZDwmxBE+IZmozwmf7hMS2rJE1q9UEaN50eIPUnotQ1G6
dDHg7t/5I/XNrtU8WyV21YR4ZnEbpUtg5R150MgFUegGcpzM1qc5LKBSGI0UdkQ
fwIDAQAB
`pragma protect control run_phase="Simulation"
`pragma protect control decryption=license_string("counter")
`pragma protect end_toolblock

`pragma protect begin
module counter (out,clk,enable,reset);
    output [7:0] out;
    input      clk, enable, reset;
    reg [7:0]   out;

```

```
always @( posedge clk )
begin
    if ( reset )
        out <= 8'b0 ;
    else if ( enable )
        out <= out + 1;
end
endmodule // counter
`pragma protect end
```

1.3 Run *ipecrypt* to generate encrypted IP

The command to encrypt the sample IP is:

```
ipecrypt --infile counter_2e.v --outfile counter_e.v
```

2. Alternative ways to add protect directives

For convenience, the *ip encrypt* allows the `protect` directives through a separate file. In this case, the whole IP will be encrypted.

2.1 YAML

The directives can be specified through a YAML file. Information supplied through the YAML file is language independent. Based on the language option specified with the command, language specific `protect` directives are added. The YAML file should have “`.yaml`” file extension.

The sample YAML file (`directives.yaml`) follows:

```
version: 3
author: IP Encrypter
author_info: "IPEncrypter.com"
data_method: aes256-cbc-hmac-sha512
commonblock:
  license_proxyenv: IPE_V3_PROXY
  license_certificate: |-  
    MIIFWzCCA6sCAQEwDQYJKoZIhvcaNAQELBQAwgamxCzAJBgNVBAYTA1VTMRMwEQQYD  
    VQQIDApDYWxpZm9ybmlhMREwDwYDVQQHDAhTYW4gSm9zZTEYMBYGA1UECgwPaXB1  
    bmNyeXB0ZXIuY29tMRYwFAYDVQQLDA1JUCBQcm90ZWN0aW9uMRUwEwYDVQQDDAxJ  
    UCBFBmNyeXB0ZXIxIzAhBgkqhkiG9w0BCQEWFGLuZm9AaXB1bmNyeXB0ZXIuY29t  
    MB4XDTIwMDQwNjA1MjAwM1oXDTMwMDQwNDA1MjAwM1owgaoxCzAJBgNVBAYTA1VT  
    MRMwEQYDVQQIDApDYWxpZm9ybmlhMREwDwYDVQQHDAhTYW4gSm9zZTEYMBYGA1UE  
    CgwPaXB1bmNyeXB0ZXIuY29tMRYwFAYDVQQLDA1JUCBQcm90ZWN0aW9uMRwwGgYD  
    VQQDBDNJCFCfbmNyeXB0ZXIgQ2xpZm50MSMwIQQJKoZIhvcaNAQkBFhRpbmZvQGlw  
    ZW5jcnlwdGVyLmNbTCCAIwDQYJKoZIhvcaNAQEBBQADggIPADCCAgcCggIBAldu  
    iGzVDVr4F7u+TSqYD4o4G0Li2mKdpgO2FXufEnidsTdwv1s8sY71lyJFO5114TYR  
    ctKCVpjliFNy6Wu2uQBwfDVnXwR/vcv7yDeHGGGdorHi1JPwpbq6MZ/fYuNJiytu  
    U9SzxMTHH3So6gv/f3AicT9xGAL8O92vJtDTiim805TxOmax4isQvEXteCfcJpGe  
    tW7Z377Crp6s0KKY98170G/SI1QAsvK1YdGmJaPQeICSnBZJp1/y8G6k3oNJGRjn  
    W+F5ZtcPh2H2MMrIMOYj6oaA2xbVaKyBzCdyccnOAqkSGTJ5+ajsLGpDdn88BfEJ5  
    G9t7juI/Eop5azYrDLvoHipe9CVHjnwl+njwrn4PAqnPdpfv9GzPfVnh+r74qR2  
    pEWLsw/8Szo90k+qsh3HO3RIUvDrnzvmLr0fxp3ljmdxe4rozdyGUbvs3jKEhyz  
    /t49Gt0L7MxRC61JY1mUhk2U+2oizPUxvBMM3C4EIc5I+11cLOJTppY4mF9oz2  
    kh++G/QUnHFip2J4GqgPB0DpBFNlch1r0JMHZehhSji+L/tpJcq3ri8+iZQbsO  
    FmY1SEru8SwT7+TETyG76H+sFIP5rtB1JLo+wBTpY8e6jPzqy/MpJT4mak/kDovG  
    XDagdhSHO2v8VDufiC5WAtMiLKQHwYRvpeD0JtjdAgMBAAEwDQYJKoZIhvcaNAQEL  
    BQADggIBAATxIpt1AMh/zpQc2EdZd8q37k14jPbXX1X3mQFTBMwGEmxmpg9ywhps  
    4drGyzZrhkC1t7p2PhruOJRnGfZNfsDKBky/YuuHyXOkgLCPFtWotDEi/coj391  
    1fWnN5oM/+fm3oWnUQ1mYDP/WoaKzp87tSHTxj/Z2pD0IBzTUuNa5lyfinCTEXXO  
    +UOrE66Bqe3JiQhRTbVAjvZ/r/m1MvkabUFaNHg95VzuU4pu1D8JLdq6SBMXk0rE  
    Q3+hGPynQc4d5Ns1P6kolADBrijBoqMyTrjTOg4/Ysbo6jJMrv1VI+USDS3poma8y  
    cMYK6hkss90i8gfmp8ZmFJej69Li1J1U+ibPY0imoS0M2nzDJ5vHm6Hmr+Zw6TkP  
    ncp6Iz3YzKV8P/Pxp+ilakDRRDA57tuABKD16bkSoRw40ZSlnJ2SR30FaF/BvHcf  
    nCkO/aDDFan/5C3qSFpseDqI34TXZeQyAOIE9qGv1rBeQIrldPlnLSkGKmqj/vp9e  
    NAkePO9smchle4Eoh8ktFM5B3VtMt4cy60YaOHGUvIzAyORErzxuuuFf+yfbWWPa  
    RmB3q2h9HV4QPtkkAy7vKUXfyCGew1TfojY2SBhQHgzOdR/pBbxD1c2AxXuKnvJ9  
    JK7gv7TJyHa7dTFOYAWCfrwAlvetx5qGHIkWaiUHXeMKK9ooR838  
  toolblock:  
    key_keyowner: ipencrypter  
    key_keyname: ipencrypter_key1  
    key_method: rsa  
    key_public_key: |-  
      MIICIJANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAZxrW73KUIII/a6khZXDvz  
      7/cgzSMdkGq2xbNqQ/gymJVFBuM+Nv1wjJcZmS7cJwLwA0LVv8EqWcy3Wqc3cceT  
      EFj1TZsmwsrcGAAprI51iOEwAOn1VPNfQT8fQJFkPdJpLB62Q8BaYvUP9KmAVCK6  
      DeCWGKYf0yBGjG411AeYxQ9VuVpIzPSO6eZoxH81EXR1s81oy60kzGG0WRbR29FO  
      YY2ti7HyWvAHStbURPZ6PhGMM62PX3xrsO2obGQbE0d52X5BAy9G6oAAUY+FP1QD  
      oR05Xryfedvan30mc1kxD+G2USeCJ6DI+fyqkQH1qyV7sISTgnEzE+wcW5phCAZ  
      k4ZcC7zItnRjbVN1NHSSiiJq3pT3s8gf9ohMI9F72aWhM3DKCNX/D41qULjsYvg
```

```

I2C8pr6ILVdE3da4Ueq9Z/9aARWGYMfd0pQ7qkieV/YVN5rWAFTNLIuVyyV19oEZ
mRnKZF3N+SEdLnPXATTi004t9svn3zDZ7WHKx7GT5/IYe6KmUZ38CHOXMUTJKQqR
zXv+5kKDRvRD1EKtYkkh12zGyUfizo50tUy2OXWo8ZfqEtGcob1GqpgOVQ0+us1/
z64cd612/RMkix2B9ETPudMEp8UNHBKz2zRnEsadUquVu+SUzfUPkTvilSvo4Hx
aSPWSriewi0V9UYKNQXuOvsCAwEAAQ==

toolblock:
key_keyowner: ipencrypter
key_keyname: ipeencrypter_key2
key_method: rsa
key_public_key: |
    MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIIBCgKCAQEa2xEbEbK+w5BWTBoOLQRB
    qyK2hYWRC2z85eEncog6pyIisXeoIFCOKfVVCCBm6u+ebKJqmVn81hmkKICJv7pb
    mO/LWJEQwyDsOqJYquTDMC+zqmqc7DuGiuyZ8/XmVpizIdtquPVDztXdnFyFZLMq
    wEBNa0zOMWMTMZjb2LOzU2jqy04vwuESAV4f1Nkl/96KZUp7pZF1X1jpNY5UoHAt
    20YSn35zdebKdNp1SKAHZDwmxBE+IZmozwmf7hMS2rJElq9UEaN50eIPUnotQ1G6
    dDHg7t/5I/XNrtU8WyV2lYR4ZnEbpUtgsR150MgFUegGcqpzM1qc5LKBSGI0UdkQ
    fwIDAQAB
control:
- run_phase: Simulation
- decryption: license_string("counter")

```

The command to encrypt the sample IP using external directive YAML file is:

```
ipecrypt --infile counter_p.v --outfile counter_e.v --directive directives.yaml -language verilog
```

2.2 JSON

The directives can be specified through a JSON file. Information supplied through the JSON file is language independent. Based on the language option specified with the command, language specific protect directives are added. The JSON file should have “.json” file extension.

The sample JSON file (directives.json) follows:

```
{
    "version": 3,
    "author": "IP Encrypter",
    "author_info": "IPEEncrypter.com",
    "data_method": "aes256-cbc-hmac-sha512",
    "commonblock": {
        "license_proxyenv": "IPE_V3_PROXY",
        "license_certificate": [
            "MIIFwzCCA6sCAQEwDQYJKoZIhvcNAQELBQAwgaMxCzAJBgNVBAYTA1VTMRMwEQUYD",
            "VQQIDApDYWxpZm9ybmlhMREwDwYDVQQDAhTYW4gSm9zZTEYMBYGA1UECgwPaXB1",
            "bmNyeXB0ZXIuY29tMRYwFAYDVQQLDA1JUCBQcm90ZWN0aW9uMRUwEwYDVQQDDAxJ",
            "UCBFbmNyeXB0ZXIxIzAhBkgkhkiG9w0BCQEWFG1uZm9AaXB1bmNyeXB0ZXIuY29t",
            "MB4XDTIwMDQwNjA1MjAwM1oXDTMwMDQwNDA1MjAwM1owgaoxCzAJBgNVBAYTA1VT",
            "MRMwEQYDVQQIDApDYWxpZm9ybmlhMREwDwYDVQQDAhTYW4gSm9zZTEYMBYGA1UE",
            "CgwPaXB1bmNyeXB0ZXIuY29tMRYwFAYDVQQLDA1JUCBQcm90ZWN0aW9uMRUwEwYDV",
            "VQQDBNJCUCFBfmNyeXB0ZXIgQ2pxZw50MSMwIQUYJKoZIhvcNAQkBFhRpbmZvQGlw",
            "ZW5jcnlwGvYlmNvbTCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgcGgIBALdu",
            "iGzVDVr4F7u+TSqYD4o4G0Li2mKdpqO2FXufEnidSTdwV1s8sY71lyJFO5114TYR",
            "ctKCVpjliFNy6Wu2uQBWFvDvnXwR/vcv7yDeHGGdorHi1JPwpbg6MZ/fYuNJiytu",
            "U9SzxtMTHH3So6gv/f3AicT9xGAL8O92vJtDTiim805TxOmax4isQvEXteCfcJpGe",
            "tW7Z377Crp6s0KKY98170G/S1lQAsvK1YdGmJaPQeICSnBZJp1/y8G6k3oNJRjn",
            "W+F5ZtcPh2H2MMrIMOYj6oaA2xbVaKyBzCdycoAqkSGTJ5+ajsLGpDdn88bfEJ5",
            "G9t7juI/Eop5azYrDLvoHipe9CVHjnwl+njwrn4PAqnPdpfv9GzPfVNh+7r74qR2",
            "pEWLsw/8SzO90k+qsh3HO3RIUvDfRnzvmLr0fxp3Ljmdxe4rozdyGUbvs3jKEhyz",
            "/t49GtOL7MxRC61JYJlmUhk2U+2oizPUXuvBMM3C4Elc5I+11cLOJTppY4mf9oz2",
            "kh++G/QUnHFip2J4GqgPB0DpBFFnLcHlr0JMHZehhSji+L/tpJcq3nri8+iZQbsO",
            "FmY1SEru8SwT7+TETyG76H+sfIP5RtB1JLO+wBtpY8e6jPzqy/MpJT4mak/kDovG",
            "XDagdhSHO2v8VDufiC5WAtMilQHwYRvpE0D0JtjDAGMAAEwDQYJKoZIhvcNAQEL",
            "X"
        ]
    }
}
```

```

        "BQADggIBAATxIpt1AMh/zpQc2EdZd8q37k14jPbXX1X3mQFTBMWgEmxmpg9ywhps",
        "4drGyzZrhcK1t7p2PhruOJRnGFZNfsDKBkY/YuuHyX0kgLCFFtWotDEi/coj391",
        "1fWnN5oM/+fm3oWnUQ1mYDP/WOaKzp87tSHTxj/Z2pD0IBzTUUnNa5lyfinCTEXXO",
        "+UOrE66Bqe3JiQhRTbVAjvz/r/m1MVkabUFaNH9g5VzuU4pu1D8JLdq6SBMXk0rE",
        "Q3+hGPynQc4d5Ns1P6kolADBkjBoqMyTrjTOg4/YSbo6jJMrv1VI+USDS3poma8y",
        "cMYK6hkss90i8gfpmp8ZmFJej69Li1J1u+ibPY0imoS0M2nzDJ5vHm6Hmr+Zw6TkP",
        "ncp6Iz3YzKV8P/Pxp+ilakDRRDA57tuABKD16bkSoRW40ZSlnj2SR30FaF/BvHCf",
        "nCkO/aDDFan/5C3qSFpseDqI34TXZeQyAOIE9qGv1rBeQIrdPlnLSkGKmqj/vp9e",
        "NAkePO9smchle4Eoh8kTFM5B3VtMt4cy60YaOHGUvIzAyORErzxuuuFf+yfbWWPa",
        "RmB3q2h9HV4QPtkkAytvkUXfyCGeWlTfOjY2SBhQHgzOdR/pBbxDlc2AxXuKnvj9",
        "JK7gv7TJyHa7dTFOYAWCfrwAlvetx5gGHIkWaiUHXeMKK9ooR838"
    ],
},
"toolblock": [
{
    "key_keyowner" : "ip encrypter",
    "key_keyname" : "ip encrypter_key1",
    "key_method" : "rsa",
    "key_public_key" : [
        "MIICIJjANBgkqhkiG9w0BAQEFAACAg8AMIICCgKCAgEAzkwR73KUII/a6khZXDvz",
        "7/cgzSMdkGq2xbNqQ/gvmJVFBuM+Nv1wjJcZmS7cJwLwA0LvV8EqWcy3Wqc3cceT",
        "EFj1TZsMswrcGAaPrI51iOEwAAon1VPNfQT8fQJFkPdJpLB62Q8BaYvUP9KmAVCK6",
        "DeCWGKYf0yBGjG411AeYxQ9VuVp1zPS06eZ0xH81EXR1S81oy60kzGG0WRbR29FO",
        "YY2ti7HyWvAHStURPZ6PhGMM62PX3xrsO2obGQbE0d52X5BAy9G6oAAUY+FP1QD",
        "oR05Xryfecvdvan3OmclkxD+G2USeCJ6DI+fyqkQH1qyV7sISTgnEzE+wCw5phCAZ",
        "k4ZC7zItnRjbVN1NHSSiiJq3pT3s8gf9ohtMI9F72aWhM3DKCNX/D41qULjsYvg",
        "I2C8pr6ILVdE3da4Ueq9Z/9aARWGyMfd0pQ7qkieV/YVN5rWAftNLiuVvyi9oEZR",
        "mRnKZF3N+SEDLnPXATTi004t9svn3zDz7WHKx7GT5/IY6KmUZ38CHOXMUTJKQqR",
        "zXv+5kKDRvRD1EKtYkkh1zGyUfizo5OtUy2OXWo8ZfqEtGcob1GqpgOVQ0+us1/",
        "z64cd612/RMkix2B9ETPudMeP8UNHBRz2zRnEsadUquVu+SUhzfUPkTv1Svo4Hx",
        "aSPWSriewi0V9UYKNQXuOvsCAwEAAQ=="
    ]
},
{
    "key_keyowner" : "ip encrypter",
    "key_keyname" : "ip encrypter_key2",
    "key_method" : "rsa",
    "key_public_key" : [
        "MIIBIjANBgkqhkiG9w0BAQEFAOCQ8AMIIBCgKCAQEA2xEbEbK+w5BWTBoOLQRB",
        "qyK2hYWRC2z85eEncog6pyIisXeoIFCOKfVVCCBm6u+ebKJqmVN81hmkKICJv7pb",
        "mO/LWJEQwyDsOqJYquTDMC+zqmqc7DuGiuYZ8/XmVpiziDtquPVDztXdnFyFZLMq",
        "wEBNa0zOMWTMZjb2LozU2jqy04vwuESAV4f1Nkl/96KZUp7pZF1X1jjpNY5UoHAt",
        "20YSn35zdebKdNp1SKAHZDwmxBE+IZmozwmf7hMS2rJE1q9UEaN5OeIPUnotQ1G6",
        "dDHg7t/5I/XNrtU8WyV21YR4ZnEbpUtgsR150MgFUegGcqpzM1qc5LKBSGI0UdkQ",
        "fwIDAQAB"
    ],
    "control" : [
        {"run_phase" : "Simulation"},
        {"decryption" : "license_string(\"counter\")"}
    ]
}
]
}
}

```

The command to encrypt the sample IP using external directive JSON file is:

```
ipecrypt --infile counter_p.v --outfile counter_e.v --directive directives.json -language verilog
```

2.3 Plain text

The directives can be specified through a text file. The text file should have “.txt” file extension.

For VHDL, use `protect directive instead of `pragma protect. For convenience, the tool can convert directives from Verilog to VHDL and vice-versa if the language option is specified with the command.

The sample directive file (directives.txt) follows:

```
`pragma protect version=3
`pragma protect author="IP Encrypter"
`pragma protect author_info="IPEncrypter.com"
`pragma protect data_method="aes256-cbc-hmac-sha512"

`pragma protect begin_commonblock
`pragma protect license_proxyenv="IPE_V3_PROXY"
`pragma protect license_certificate
MIIFWzCCA6sCAQEwDQYJKoZIhvcNAQELBQAwgAxCzAJBgNVBAYTA1VTMRMwEQYD
VQQIDApDYWxpZm9ybmlhMREwDwYDVQQHDAhTYW4gSm9zZTEYMBGA1UECgwPaXB1
bmNyeXB0ZXIuY29tMRYwFAYDVQQLDA1JUCBQcm90ZWN0aW9uMRUwEwYDVQQDAXj
UCBFbmNyeXB0ZXIxIzAhBgkqhkiG9w0BCQEWFGluZm9AaXBlbmNyeXB0ZXIuY29t
MB4XDTIwMDQwNjA1MjAwM1oXDMwMDQwNDA1MjAwM1owgaoxCzAJBgNVBAYTA1VT
MRMwEQYDVQQIDApDYWxpZm9ybmlhMREwDwYDVQQHDAhTYW4gSm9zZTEYMBGA1UE
CgwPaXB1bmNyeXB0ZXIuY29tMRYwFAYDVQQLDA1JUCBQcm90ZWN0aW9uMRwwGgYD
VQQDBNJUCBFbmNyeXB0ZXIxQ2xpZW50MSMwIQQJKoZIhvcNAQkBFhRpbmZvQGlw
ZW5jcnlwdGVyLmNvbTCCAI1wDQYJKoZIhvcNAQEBBQADggIPADCCAgcCggIBALdu
iGzVDVr4F7u+TSqYD4o4G0Li2mKdpgO2FXufEnidSTdwVls8sY71lyJFC05114TYR
ctKCvpjliFNY6Wu2uQBWFdVnXwR/vcv7yDeHGGGdorHi1JPwpbq6MZ/fYuNJiytu
U9SzxMTHH3S06gv/f3Aict9xGAL8092vJTDtiim805Tx0max4isQvExteCfcJpGe
tw7z377Crp6sOKKY98170G/SI1QAsvK1YdGmJaPQeICSnBZJp1/y8G6k3oNJGRjn
W+F5ZtcPh2H2MMrIMOYj6oaA2xbVaKyBzCdycnOAqkSGTJ5+ajsLGpDdn88BfEJ5
G9t7juI/Eop5azYrDLvoHipe9CVHjnwl+njwrn4PAqnPdpfv9GzPfVNh+7r74qR2
pEWlsw/8Sz090k+qsh3HO3RIuVdfRnzvmLr0fxp3Ljmdxe4rozdyGUbvs3jKEhz
/t49GtOL7MxRC61JYJlmUhk2U+2oizPUXuvBMM3C4Eic5I+11cLOJTppY4mF9oz2
kh++G/QUnHFip2J4GqgPB0DpBFFnLch1r0JMHZehhSji+l/tpJcq3nri8+iZQbsO
FmY1SEru8Sw7+TETyG76H+sfp5RtB1JLO+wBTpY8e6jPzqy/MpJT4mak/kDovG
XDaghSHO2v8VDufiC5WAtMlkQHwYRvpeD0JtjDAGMBAEwDQYJKoZIhvcNAQEL
BQADggIBAAIxIpt1AMh/zpQc2Edzd8q37k14jPbxX1X3mQFTBMWgEmxmpg9ywhps
4drGyzRhcK1t7p2PhruOJRnGfZNfsDkBy/YuuHxOkgLCFtWotDEi/coj391
1fWnN5oM/+fm3oWnQ1mYDP/WoAkzp87tSHTxj/Z2pD0BzTUUnNa5lyfinCTEXXO
+UOrE66Bqe3JiQhRTbVAjvZ/r/m1MVkabUFaNH9g5VzuU4pu1D8JLdq6SBMXk0rE
Q3+hGPynQc4d5Ns1P6k01ADBjBoqMyTrjTOg4/YSbo6jJMrv1VI+USDS3poma8y
cMYK6hkss90i8gfmp8ZmFjej69Li1J1U+ibPY0iimoSM0n2zD5vHm6Hmr+Zw6TkP
ncp6Iz3YzKV8P/Pxp+ilakDrrda57tuABKD16bkSoRW40ZslnJ2SR30FaF/BvHcf
ncko/aDDFan/5C3qSFpseDqI34TXZeQyAOIE9qGv1rBeQIrdPlnLSkGKmqj/vp9e
NAkePO9smchle4Eoh8kTFM3VtMt4cy60YaOHGUvIzAyORErzxuuuFf+yfbWWPa
RmB3q2h9HV4QPtkkAytVkuXfyCGeW1TfojY2SBhQhgzOdR/pBbxD1c2AxXuKnvj9
JK7gv7TJyHa7dTFOYAWCfrwAlvetx5qGHIkWaiUHXeMKK9ooR838
`pragma protect end_commonblock

`pragma protect begin_toolblock
`pragma protect key_keyowner = "ip encrypter"
`pragma protect key_keyname = "ip encrypter_key1"
`pragma protect key_method = "rsa"
`pragma protect key_public_key
MIICIJANBqkqhkiG9w0BAQEFAAOCAg8AMIIICCgKCAGEAzwxR73KUII/a6khZxDvz
```

```

7/cgzSMdkGq2xbNqQ/gvmJVFBuM+Nv1wjJcZmS7cJwLwA0LVv8EqWcy3Wqc3cceT
EFj1TZsMswwwGAApRI51iOEwAOOn1VPMFQT8fQJFkPdJpLB62Q8BaYvUP9KmAVCK6
DeCWGKYf0yBGjG411AeYxQ9VuVpIzPSO6eZOxH81EXR1S81oy60kzGG0WRbR29FO
YY2ti7HyWvAHStURPZ6PhGMM62PX3xrsO2obGQbE0d52X5BAy9G6oAAUY+FPlQD
oR05Xryfecdvan3OmclkxD+G2USeCJ6DI+fyqkQH1qyV7sISTgnEzE+wcW5phCAZ
k4ZCC7zItnRJbVN1NHSSiiJq3pT3s8gf9htMI9F72aWhM3DKCNX/D41qULjsYvg
I2C8pr6ILVdE3da4Ueq9Z/9aARWGYMfd0pQ7qkieV/YVN5rWAFTNLiuVyyI9oEZRM
mRnKZF3N+SEdLnPXATTi004t9svn3zDZ7WHKx7GT5/IYe6KmUZ38CHOXMUTJKQqR
zXv+5kKDRvRD1EKtYkkh12zGyUfizo50tUy2OXwo8ZfqEtGcob1GqpgOVQ0+us1/
z64cd612/RMKix2B9ETPudMEp8UNHBKz2zRnEsadUquVu+SUunzfUPkTv1lSvo4HxaSPWSriewi0V9UYKNQXuOvsCAwEAAQ==

`pragma protect end_toolblock

`pragma protect begin_toolblock
`pragma protect key_keyowner = "ip encrypter"
`pragma protect key_keyname = "ip encrypter_key2"
`pragma protect key_method = "rsa"
`pragma protect key_public_key
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIIBCgKCAQEA2xEbEbK+w5BWTBoOLQRB
qyK2hYWRC2z85eEncog6pyIisXeoIFCOKfVVCCBm6u+ebKJqmvN81hmkKICJv7pb
mO/LWJEQwyDsOqJYquTDMC+zqmqc7DuGiuyZ8/XmVpiziDtqquPVDztXdnFyFZLMq
wEBNa0zOMWTM2jb2LOzU2jqy04vwuESAV4f1Nk1/96KZUp7pZF1X1jjpNY5UoHAt
20YSn35zdebKdNp1SKAHZDwmxBE+IZmozwmf7hMS2rJE1q9UEaN5OeIPUnotQ1G6
dDHg7t/5I/XNrtU8WyV21YR4ZnEbpUtgsR150MgFUegGcpzM1qc5LKBSGI0UdkQ
fwIDAQAB
`pragma protect control run_phase="Simulation"
`pragma protect control decryption=license_string("counter")
`pragma protect end_toolblock

```

The command to encrypt the sample IP using external directive text file is:

```
ipecrypt --infile counter_p.v --outfile counter_e.v --directive directives.txt
```