IPENCRYPTER.COM

Encrypt HDL Intellectual Property (IP)

ipecrypt

compliant with
IEEE Std 1735™-2023 standard

IP Encrypter

User Guide Version 23.1.0

Table of Contents

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://ipencrypter.com/downloads/hdl-ip-encrypter-tools.

For license proxy and decryption products conforming to the “IEEE Std 1735™-2023” visit:
https://ipencrypter.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
// ipencrypter.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
// ipencrypter.com
//***********************************
`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
MIIFwzCCA6sCAQEwDQYJKoZIhvcNAQELBQAwgaMxCzAJBgNVBAYTAlVTMRMwEQYD
VQQIDApDYWxpZm9ybmlhMREwDwYDVQQHDAhTYW4gSm9zZTEYMBYGA1UECgwPaXBl
bmNyeXB0ZXIuY29tMRYwFAYDVQQLDA1JUCBQcm90ZWN0aW9uMRUwEwYDVQQDDAxJ
UCBFbmNyeXB0ZXIxIzAhBgkqhkiG9w0BCQEWFGluZm9AaXBlbmNyeXB0ZXIuY29t
MB4XDTIwMDQwNjA1MjAwM1oXDTMwMDQwNDA1MjAwM1owgaoxCzAJBgNVBAYTAlVT
MRMwEQYDVQQIDApDYWxpZm9ybmlhMREwDwYDVQQHDAhTYW4gSm9zZTEYMBYGA1UE
CgwPaXBlbmNyeXB0ZXIuY29tMRYwFAYDVQQLDA1JUCBQcm90ZWN0aW9uMRwwGgYD
VQQDDBNJUCBFbmNyeXB0ZXIgQ2xpZW50MSMwIQYJKoZIhvcNAQkBFhRpbmZvQGlw
ZW5jcnlwdGVyLmNvbTCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBALdu
iGzVDVr4F7u+TSqYD4o4G0Li2mKdpgO2FXufEnidSTdwV1s8sY7llyJFO5114TYR
ctKCVpjliFNY6Wu2uQBWfDVnXwR/vcv7yDeHGGGdorHi1JPwpbq6MZ/fYuNJiytu
```

```
U9SzxMTHH3So6gv/f3AicT9xGAL8O92vJtDTiim805TxOmax4isQvEXteCfcJpGe
tW7Z377Crp6s0KKY98170G/SIlQAsvK1YdGmJaPQeICSnBZJp1/y8G6k3oNJGRjn
W+F5ZtcPh2H2MMrIMOYj6oaA2xbVaKyBzCdycnOAqkSGTJ5+ajsLGpDdn88BfEJ5
G9t7juI/Eop5azYrDLvoHipe9CVHjnwl+njwrn4PAqnPdpfv9GzPfVNh+7r74qR2
pEWLsw/8SzO90k+qsh3HO3RIUvDfRnzvmLr0fxp3Ljmdxe4rozdyGUbvs3jKEhyz
/t49GtOL7MxRC6lJYJlmUHk2U+2oizPUXuvBMM3C4EIc5I+11cLOJTppY4mF9oz2
kh++G/QUnHFip2J4GqgPBoDpBFFnLcHlr0JMHZehhSji+L/tpJcq3nri8+iZQbsO
FmY1SEru8SWt7+TETyG76H+sfIP5RtB1JLO+wBTpY8e6jPzqy/MpJT4mak/kDovG
XDagdhSHO2v8VDufiC5WAtMiLKQHwYRvpeD0JtjDAgMBAAEwDQYJKoZIhvcNAQEL
BQADggIBAATxIpt1AMh/zpQc2EdZd8q37kl4jPbXXlX3mQFTBMWgEmxmpg9ywhps
4drGyzZrhcK11t7p2PhruOJRnGfZNfsDKBkY/YuuHyXOkgLCPFtWotDEi/coj391
lfWnN5oM/+fm3oWnUQlmYDP/WOaKzp87tSHTxj/Z2pD0IBzTUnNa5lyfinCTEXXO
+UOrE66Bqe3JiQhRTbVAjvZ/r/m1MVkabUFaNH9g5VZuU4pu1D8JLdq6SBMXk0rE
Q3+hGPynQc4d5NslP6kolADBRjBoqMyTrjTOg4/YSbo6jJMrvlVI+USDS3poma8y
cMYK6hkss90i8gfmp8ZmFJej69Li1JlU+ibPY0imoS0M2nzDJ5vHm6Hmr+Zw6TkP
ncp6Iz3YzKV8P/Pxp+ilakDRRDA57tuABKD16bkSoRW40ZSlnJ2SR30FaF/BvHCf
nCkO/aDDFan/5C3qSFpseDqI34TXZeQyAOIE9qGv1rBeQIrdPlnLSkGKmqj/vp9e
NAkePO9smchle4Eoh8kTFM5B3VtMt4cy60YaOHGUvIzAyORErzxuuuFf+yfbWWPa
RmB3q2h9HV4QPtkkAytVkUXfyCGeWlTfOjY2SBhQHgzOdR/pBbxDlc2AxXuKnvj9
JK7gv7TJyHa7dTF0YAWCfrwAlvetx5qGHIkWaiUHXeMKK9ooR838
`pragma protect end_commonblock

`pragma protect begin_toolblock
`pragma protect key_keyowner = "ipencrypter"
`pragma protect key_keyname = "ipencrypter_key1"
`pragma protect key_method = "rsa"
`pragma protect key_public_key
MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAzxwR73KUII/a6khZXDvz
7/cgzSMdkGq2xbNqQ/gvmJVFBuM+Nv1wjJcZmS7cJwLwA0LVv8EqWcy3Wqc3cceT
EFjlTZsMswrcGAaPrI51iOEwAOnlVPNfQT8fQJFkPdJpLB62Q8BaYvUP9KmAVCK6
DeCWGKYf0yBGjG411AeYxQ9VuVpIzPSO6eZOxH81EXRlS8loy60kzGG0WRbR29FO
YY2ti7HyWvAHSbtURPZ6PhGMM62PX3xrsO2obGQbE0d52X5BAy9G6oAAUY+FPlQD
oR05Xryfecdvan3OmclkxD+G2USeCJ6DI+fyqkQH1qyV7sISTgnEzE+wcW5phCAZ
k4ZcC7zItnRJbVN1NHSSiiJq3pT3s8gf9ohtMI9F72aWhM3DKCNX/D41qULjsYvg
I2C8pr6ILVdE3da4Ueq9Z/9aARWGyMfd0pQ7qkieV/YVN5rWAFtNLIuVyvI9oEZR
mRnKZF3N+SEdLnPXATTiO04t9svn3zDZ7WHKx7GT5/IYe6KmUZ38CHOXMUTJKQqR
zXv+5kKDRvRDlEKtYkkhl2zGyUfizo5OtUy2OXWo8ZfqEtGcob1GqpgOVQ0+us1/
z64cd612/RMkix2B9ETPudMEp8UNHBKz2zRnEsadUquVu+SUnzfUPkTvilSvo4Hx
aSPWSriewi0V9UYKNQXuOvsCAwEAAQ==
`pragma protect end_toolblock

`pragma protect begin_toolblock
`pragma protect key_keyowner = "ipencrypter"
`pragma protect key_keyname = "ipencrypter_key2"
`pragma protect key_method = "rsa"
`pragma protect key_public_key
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA2xEbEbK+w5BWTBoOLQRB
qyK2hYWRC2z85eEncog6pyIisXeoIFCOKfVVCCBm6u+ebKJqmvN8lhmkKICJv7pb
mO/LWJEQwyDsOqJYquTDMC+zqmqc7DuGiuYZ8/XmVpiziDtquPVDztXdnFyFZLMq
wEBNa0zOMWTMZjb2LOzU2jqy04vwuESAV4f1Nkl/96KZUp7pZF1X1jjpNY5UoHAt
20YSn35zdebKdNp1SKAHZDwmxBE+IZmozwmf7hMS2rJElq9UEaN5OeIPUnotQlG6
dDHg7t/5I/XNrtU8WyV2lYR4ZnEbpUtgsRl5OMgFUegGcqpzM1qc5LKBSGI0UdkQ
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 *ipecrypt* 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: |-
    MIIFwzCCA6sCAQEwDQYJKoZIhvcNAQELBQAwgaMxCzAJBgNVBAYTAlVTMRMwEQYD
    VQQIDApDYWxpZm9ybmlhMREwDwYDVQQHDAhTYW4gSm9zZTEYMBYGA1UECgwPaXBl
    bmNyeXB0ZXIuY29tMRYwFAYDVQQLDA1JUCBQcm90ZWN0aW9uMRUwEwYDVQQDDAxJ
    UCBFbmNyeXB0ZXIxIzAhBgkqhkiG9w0BCQEWFGluZm9AaXBlbmNyeXB0ZXIuY29t
    MB4XDTIwMDQwNjA1MjAwM1oXDTMwMDQwNDA1MjAwM1owgaoxCzAJBgNVBAYTAlVT
    MRMwEQYDVQQIDApDYWxpZm9ybmlhMREwDwYDVQQHDAhTYW4gSm9zZTEYMBYGA1UE
    CgwPaXBlbmNyeXB0ZXIuY29tMRYwFAYDVQQLDA1JUCBQcm90ZWN0aW9uMRwwGgYD
    VQQDDBNJUCBFbmNyeXB0ZXIgQ2xpZW50MSMwIQYJKoZIhvcNAQkBFhRpbmZvQGlw
    ZW5jcnlwdGVyLmNvbTCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBALdu
    iGzVDVr4F7u+TSqYD4o4G0Li2mKdpgO2FXufEnidSTdwV1s8sY7llyJFO5114TYR
    ctKCVpjliFNY6Wu2uQBWfDVnXwR/vcv7yDeHGGGdorHi1JPwpbq6MZ/fYuNJiytu
    U9SzxMTHH3So6gv/f3AicT9xGAL8O92vJtDTiim805TxOmax4isQvEXteCfcJpGe
    tW7Z377Crp6s0KKY98170G/SIlQAsvK1YdGmJaPQeICSnBZJp1/y8G6k3oNJGRjn
    W+F5ZtcPh2H2MMrIMOYj6oaA2xbVaKyBzCdycnOAqkSGTJ5+ajsLGpDdn88BfEJ5
    G9t7juI/Eop5azYrDLvoHipe9CVHjnwl+njwrn4PAqnPdpfv9GzPfVNh+7r74qR2
    pEWLsw/8SzO90k+qsh3HO3RIUvDfRnzvmLr0fxp3Ljmdxe4rozdyGUbvs3jKEhyz
    /t49GtOL7MxRC6lJYJlmUHk2U+2oizPUXuvBMM3C4EIc5I+11cLOJTppY4mF9oz2
    kh++G/QUnHFip2J4GqgPBoDpBFFnLcHlr0JMHZehhSji+L/tpJcq3nri8+iZQbsO
    FmY1SEru8SWt7+TETyG76H+sfIP5RtB1JLO+wBTpY8e6jPzqy/MpJT4mak/kDovG
    XDagdhSHO2v8VDufiC5WAtMiLKQHwYRvpeD0JtjDAgMBAAEwDQYJKoZIhvcNAQEL
    BQADggIBAATxIpt1AMh/zpQc2EdZd8q37kl4jPbXXlX3mQFTBMWgEmxmpg9ywhps
    4drGyzZrhcK11t7p2PhruOJRnGfZNfsDKBkY/YuuHyXOkgLCPFtWotDEi/coj391
    lfWnN5oM/+fm3oWnUQlmYDP/WOaKzp87tSHTxj/Z2pD0IBzTUnNa5lyfinCTEXXO
    +UOrE66Bqe3JiQhRTbVAjvZ/r/m1MVkabUFaNH9g5VZuU4pu1D8JLdq6SBMXk0rE
    Q3+hGPynQc4d5NslP6kolADBRjBoqMyTrjTOg4/YSbo6jJMrvlVI+USDS3poma8y
    cMYK6hkss90i8gfmp8ZmFJej69Li1JlU+ibPY0imoS0M2nzDJ5vHm6Hmr+Zw6TkP
    ncp6Iz3YzKV8P/Pxp+ilakDRRDA57tuABKD16bkSoRW40ZSlnJ2SR30FaF/BvHCf
    nCkO/aDDFan/5C3qSFpseDqI34TXZeQyAOIE9qGv1rBeQIrdPlnLSkGKmqj/vp9e
    NAkePO9smchle4Eoh8kTFM5B3VtMt4cy60YaOHGUvIzAyORErzxuuuFf+yfbWWPa
    RmB3q2h9HV4QPtkkAytVkUXfyCGeWlTfOjY2SBhQHgzOdR/pBbxDlc2AxXuKnvj9
    JK7gv7TJyHa7dTF0YAWCfrwAlvetx5qGHIkWaiUHXeMKK9ooR838
toolblock:
  key_keyowner: ipencrypter
  key_keyname: ipencrypter_key1
  key_method: rsa
  key_public_key: |-
    MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAzxwR73KUII/a6khZXDvz
    7/cgzSMdkGq2xbNqQ/gvmJVFBuM+Nv1wjJcZmS7cJwLwA0LVv8EqWcy3Wqc3cceT
    EFjlTZsMswrcGAaPrI51iOEwAOnlVPNfQT8fQJFkPdJpLB62Q8BaYvUP9KmAVCK6
    DeCWGKYf0yBGjG411AeYxQ9VuVpIzPSO6eZOxH81EXRlS8loy60kzGG0WRbR29FO
    YY2ti7HyWvAHSbtURPZ6PhGMM62PX3xrsO2obGQbE0d52X5BAy9G6oAAUY+FPlQD
    oR05Xryfecdvan3OmclkxD+G2USeCJ6DI+fyqkQH1qyV7sISTgnEzE+wcW5phCAZ
    k4ZcC7zItnRJbVN1NHSSiiJq3pT3s8gf9ohtMI9F72aWhM3DKCNX/D41qULjsYvg
    I2C8pr6ILVdE3da4Ueq9Z/9aARWGyMfd0pQ7qkieV/YVN5rWAFtNLIuVyvI9oEZR
    mRnKZF3N+SEdLnPXATTiO04t9svn3zDZ7WHKx7GT5/IYe6KmUZ38CHOXMUTJKQqR
    zXv+5kKDRvRDlEKtYkkhl2zGyUfizo5OtUy2OXWo8ZfqEtGcob1GqpgOVQ0+us1/
    z64cd612/RMkix2B9ETPudMEp8UNHBKz2zRnEsadUquVu+SUnzfUPkTvilSvo4Hx
    aSPWSriewi0V9UYKNQXuOvsCAwEAAQ==
toolblock:
  key_keyowner: ipencrypter
  key_keyname: ipeencrypter_key2
  key_method: rsa
  key_public_key: |-
    MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA2xEbEbK+w5BWTBoOLQRB
    qyK2hYWRC2z85eEncog6pyIisXeoIFCOKfVVCCBm6u+ebKJqmvN8lhmkKICJv7pb
    mO/LWJEQwyDsOqJYquTDMC+zqmqc7DuGiuYZ8/XmVpiziDtquPVDztXdnFyFZLMq
    wEBNa0zOMWTMZjb2LOzU2jqy04vwuESAV4f1Nkl/96KZUp7pZF1X1jjpNY5UoHAt
    20YSn35zdebKdNp1SKAHZDwmxBE+IZmozwmf7hMS2rJElq9UEaN5OeIPUnotQlG6
    dDHg7t/5I/XNrtU8WyV2lYR4ZnEbpUtgsRl5OMgFUegGcqpzM1qc5LKBSGI0UdkQ
    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": "IPEncrypter.com",
    "data_method": "aes256-cbc-hmac-sha512",
    "commonblock": {
        "license_proxyenv":"IPE_V3_PROXY",
        "license_certificate": [
                "MIIFwzCCA6sCAQEwDQYJKoZIhvcNAQELBQAwgaMxCzAJBgNVBAYTAlVTMRMwEQYD",
                "VQQIDApDYWxpZm9ybmlhMREwDwYDVQQHDAhTYW4gSm9zZTEYMBYGA1UECgwPaXBl",
                "bmNyeXB0ZXIuY29tMRYwFAYDVQQLDA1JUCBQcm90ZWN0aW9uMRUwEwYDVQQDDAxJ",
                "UCBFbmNyeXB0ZXIxIzAhBgkqhkiG9w0BCQEWFGluZm9AaXBlbmNyeXB0ZXIuY29t",
                "MB4XDTIwMDQwNjA1MjAwM1oXDTMwMDQwNDA1MjAwM1owgaoxCzAJBgNVBAYTAlVT",
                "MRMwEQYDVQQIDApDYWxpZm9ybmlhMREwDwYDVQQHDAhTYW4gSm9zZTEYMBYGA1UE",
                "CgwPaXBlbmNyeXB0ZXIuY29tMRYwFAYDVQQLDA1JUCBQcm90ZWN0aW9uMRwwGgYD",
                "VQQDDBNJUCBFbmNyeXB0ZXIgQ2xpZW50MSMwIQYJKoZIhvcNAQkBFhRpbmZvQGlw",
                "ZW5jcnlwdGVyLmNvbTCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBALdu",
                "iGzVDVr4F7u+TSqYD4o4G0Li2mKdpgO2FXufEnidSTdwV1s8sY7llyJFO5114TYR",
                "ctKCVpjliFNY6Wu2uQBWfDVnXwR/vcv7yDeHGGGdorHi1JPwpbq6MZ/fYuNJiytu",
                "U9SzxMTHH3So6gv/f3AicT9xGAL8O92vJtDTiim805TxOmax4isQvEXteCfcJpGe",
                "tW7Z377Crp6s0KKY98170G/SIlQAsvK1YdGmJaPQeICSnBZJp1/y8G6k3oNJGRjn",
                "W+F5ZtcPh2H2MMrIMOYj6oaA2xbVaKyBzCdycnOAqkSGTJ5+ajsLGpDdn88BfEJ5",
                "G9t7juI/Eop5azYrDLvoHipe9CVHjnwl+njwrn4PAqnPdpfv9GzPfVNh+7r74qR2",
                "pEWLsw/8SzO90k+qsh3HO3RIUvDfRnzvmLr0fxp3Ljmdxe4rozdyGUbvs3jKEhyz",
                "/t49GtOL7MxRC6lJYJlmUHk2U+2oizPUXuvBMM3C4EIc5I+11cLOJTppY4mF9oz2",
                "kh++G/QUnHFip2J4GqgPBoDpBFFnLcHlr0JMHZehhSji+L/tpJcq3nri8+iZQbsO",
                "FmY1SEru8SWt7+TETyG76H+sfIP5RtB1JLO+wBTpY8e6jPzqy/MpJT4mak/kDovG",
                "XDagdhSHO2v8VDufiC5WAtMiLKQHwYRvpeD0JtjDAgMBAAEwDQYJKoZIhvcNAQEL",
                "BQADggIBAATxIpt1AMh/zpQc2EdZd8q37kl4jPbXXlX3mQFTBMWgEmxmpg9ywhps",
                "4drGyzZrhcK11t7p2PhruOJRnGfZNfsDKBkY/YuuHyXOkgLCPFtWotDEi/coj391",
                "lfWnN5oM/+fm3oWnUQlmYDP/WOaKzp87tSHTxj/Z2pD0IBzTUnNa5lyfinCTEXXO",
                "+UOrE66Bqe3JiQhRTbVAjvZ/r/m1MVkabUFaNH9g5VZuU4pu1D8JLdq6SBMXk0rE",
                "Q3+hGPynQc4d5NslP6kolADBRjBoqMyTrjTOg4/YSbo6jJMrvlVI+USDS3poma8y",
                "cMYK6hkss90i8gfmp8ZmFJej69Li1JlU+ibPY0imoS0M2nzDJ5vHm6Hmr+Zw6TkP",
                "ncp6Iz3YzKV8P/Pxp+ilakDRRDA57tuABKD16bkSoRW40ZSlnJ2SR30FaF/BvHCf",
                "nCkO/aDDFan/5C3qSFpseDqI34TXZeQyAOIE9qGv1rBeQIrdPlnLSkGKmqj/vp9e",
                "NAkePO9smchle4Eoh8kTFM5B3VtMt4cy60YaOHGUvIzAyORErzxuuuFf+yfbWWPa",
                "RmB3q2h9HV4QPtkkAytVkUXfyCGeWlTfOjY2SBhQHgzOdR/pBbxDlc2AxXuKnvj9",
                "JK7gv7TJyHa7dTF0YAWCfrwAlvetx5qGHIkWaiUHXeMKK9ooR838"
        ]
    },
    "toolblock":[
        {
            "key_keyowner" : "ipencrypter",
            "key_keyname" : "ipencrypter_key1",
            "key_method" : "rsa",
            "key_public_key" : [
                "MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAzxwR73KUII/a6khZXDvz",
                "7/cgzSMdkGq2xbNqQ/gvmJVFBuM+Nv1wjJcZmS7cJwLwA0LVv8EqWcy3Wqc3cceT",
                "EFjlTZsMswrcGAaPrI51iOEwAOnlVPNfQT8fQJFkPdJpLB62Q8BaYvUP9KmAVCK6",
                "DeCWGKYf0yBGjG411AeYxQ9VuVpIzPSO6eZOxH81EXRlS8loy60kzGG0WRbR29FO",
                "YY2ti7HyWvAHSbtURPZ6PhGMM62PX3xrsO2obGQbE0d52X5BAy9G6oAAUY+FPlQD",
                "oR05Xryfecdvan3OmclkxD+G2USeCJ6DI+fyqkQH1qyV7sISTgnEzE+wcW5phCAZ",
                "k4ZcC7zItnRJbVN1NHSSiiJq3pT3s8gf9ohtMI9F72aWhM3DKCNX/D41qULjsYvg",
                "I2C8pr6ILVdE3da4Ueq9Z/9aARWGyMfd0pQ7qkieV/YVN5rWAFtNLIuVyvI9oEZR",
                "mRnKZF3N+SEdLnPXATTiO04t9svn3zDZ7WHKx7GT5/IYe6KmUZ38CHOXMUTJKQqR",
                "zXv+5kKDRvRDlEKtYkkhl2zGyUfizo5OtUy2OXWo8ZfqEtGcob1GqpgOVQ0+us1/",
                "z64cd612/RMkix2B9ETPudMEp8UNHBKz2zRnEsadUquVu+SUnzfUPkTvilSvo4Hx",
                "aSPWSriewi0V9UYKNQXuOvsCAwEAAQ=="
            ]
        },
        {
            "key_keyowner" : "ipencrypter",
            "key_keyname" : "ipencrypter_key2",
            "key_method" : "rsa",
            "key_public_key" : [
                "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA2xEbEbK+w5BWTBoOLQRB",
                "qyK2hYWRC2z85eEncog6pyIisXeoIFCOKfVVCCBm6u+ebKJqmvN8lhmkKICJv7pb",
                "mO/LWJEQwyDsOqJYquTDMC+zqmqc7DuGiuYZ8/XmVpiziDtquPVDztXdnFyFZLMq",
                "wEBNa0zOMWTMZjb2LOzU2jqy04vwuESAV4f1Nkl/96KZUp7pZF1X1jjpNY5UoHAt",
                "20YSn35zdebKdNp1SKAHZDwmxBE+IZmozwmf7hMS2rJElq9UEaN5OeIPUnotQlG6",
                "dDHg7t/5I/XNrtU8WyV2lYR4ZnEbpUtgsRl5OMgFUegGcqpzM1qc5LKBSGI0UdkQ",
                "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
MIIFwzCCA6sCAQEwDQYJKoZIhvcNAQELBQAwgaMxCzAJBgNVBAYTAlVTMRMwEQYD
VQQIDApDYWxpZm9ybmlhMREwDwYDVQQHDAhTYW4gSm9zZTEYMBYGA1UECgwPaXBl
bmNyeXB0ZXIuY29tMRYwFAYDVQQLDA1JUCBQcm90ZWN0aW9uMRUwEwYDVQQDDAxJ
UCBFbmNyeXB0ZXIxIzAhBgkqhkiG9w0BCQEWFGluZm9AaXBlbmNyeXB0ZXIuY29t
MB4XDTIwMDQwNjA1MjAwM1oXDTMwMDQwNDA1MjAwM1owgaoxCzAJBgNVBAYTAlVT
MRMwEQYDVQQIDApDYWxpZm9ybmlhMREwDwYDVQQHDAhTYW4gSm9zZTEYMBYGA1UE
CgwPaXBlbmNyeXB0ZXIuY29tMRYwFAYDVQQLDA1JUCBQcm90ZWN0aW9uMRwwGgYD
VQQDDBNJUCBFbmNyeXB0ZXIgQ2xpZW50MSMwIQYJKoZIhvcNAQkBFhRpbmZvQGlw
ZW5jcnlwdGVyLmNvbTCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBALdu
iGzVDVr4F7u+TSqYD4o4G0Li2mKdpgO2FXufEnidSTdwV1s8sY7llyJFO5114TYR
ctKCVpjliFNY6Wu2uQBWfDVnXwR/vcv7yDeHGGGdorHi1JPwpbq6MZ/fYuNJiytu
U9SzxMTHH3So6gv/f3AicT9xGAL8O92vJtDTiim805TxOmax4isQvEXteCfcJpGe
tW7Z377Crp6s0KKY98170G/SIlQAsvK1YdGmJaPQeICSnBZJp1/y8G6k3oNJGRjn
W+F5ZtcPh2H2MMrIMOYj6oaA2xbVaKyBzCdycnOAqkSGTJ5+ajsLGpDdn88BfEJ5
G9t7juI/Eop5azYrDLvoHipe9CVHjnwl+njwrn4PAqnPdpfv9GzPfVNh+7r74qR2
pEWLsw/8SzO90k+qsh3HO3RIUvDfRnzvmLr0fxp3Ljmdxe4rozdyGUbvs3jKEhyz
/t49GtOL7MxRC6lJYJlmUHk2U+2oizPUXuvBMM3C4EIc5I+11cLOJTppY4mF9oz2
kh++G/QUnHFip2J4GqgPBoDpBFFnLcHlr0JMHZehhSji+L/tpJcq3nri8+iZQbsO
FmY1SEru8SWt7+TETyG76H+sfIP5RtB1JLO+wBTpY8e6jPzqy/MpJT4mak/kDovG
XDagdhSHO2v8VDufiC5WAtMiLKQHwYRvpeD0JtjDAgMBAAEwDQYJKoZIhvcNAQEL
BQADggIBAATxIpt1AMh/zpQc2EdZd8q37kl4jPbXXlX3mQFTBMWgEmxmpg9ywhps
4drGyzZrhcK11t7p2PhruOJRnGfZNfsDKBkY/YuuHyXOkgLCPFtWotDEi/coj391
lfWnN5oM/+fm3oWnUQlmYDP/WOaKzp87tSHTxj/Z2pD0IBzTUnNa5lyfinCTEXXO
+UOrE66Bqe3JiQhRTbVAjvZ/r/m1MVkabUFaNH9g5VZuU4pu1D8JLdq6SBMXk0rE
Q3+hGPynQc4d5NslP6kolADBRjBoqMyTrjTOg4/YSbo6jJMrvlVI+USDS3poma8y
cMYK6hkss90i8gfmp8ZmFJej69Li1JlU+ibPY0imoS0M2nzDJ5vHm6Hmr+Zw6TkP
ncp6Iz3YzKV8P/Pxp+ilakDRRDA57tuABKD16bkSoRW40ZSlnJ2SR30FaF/BvHCf
nCkO/aDDFan/5C3qSFpseDqI34TXZeQyAOIE9qGv1rBeQIrdPlnLSkGKmqj/vp9e
NAkePO9smchle4Eoh8kTFM5B3VtMt4cy60YaOHGUvIzAyORErzxuuuFf+yfbWWPa
RmB3q2h9HV4QPtkkAytVkUXfyCGeWlTfOjY2SBhQHgzOdR/pBbxDlc2AxXuKnvj9
JK7gv7TJyHa7dTF0YAWCfrwAlvetx5qGHIkWaiUHXeMKK9ooR838
`pragma protect end_commonblock

`pragma protect begin_toolblock
`pragma protect key_keyowner = "ipencrypter"
`pragma protect key_keyname = "ipencrypter_key1"
`pragma protect key_method = "rsa"
`pragma protect key_public_key
MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAzxwR73KUII/a6khZXDvz
```

```
7/cgzSMdkGq2xbNqQ/gvmJVFBuM+Nv1wjJcZmS7cJwLwA0LVv8EqWcy3Wqc3cceT
EFjlTZsMswrcGAaPrI51iOEwAOnlVPNfQT8fQJFkPdJpLB62Q8BaYvUP9KmAVCK6
DeCWGKYf0yBGjG411AeYxQ9VuVpIzPSO6eZOxH81EXRlS8loy60kzGG0WRbR29FO
YY2ti7HyWvAHSbtURPZ6PhGMM62PX3xrsO2obGQbE0d52X5BAy9G6oAAUY+FPlQD
oR05Xryfecdvan3OmclkxD+G2USeCJ6DI+fyqkQH1qyV7sISTgnEzE+wcW5phCAZ
k4ZcC7zItnRJbVN1NHSSiiJq3pT3s8gf9ohtMI9F72aWhM3DKCNX/D41qULjsYvg
I2C8pr6ILVdE3da4Ueq9Z/9aARWGyMfd0pQ7qkieV/YVN5rWAFtNLIuVyvI9oEZR
mRnKZF3N+SEdLnPXATTiO04t9svn3zDZ7WHKx7GT5/IYe6KmUZ38CHOXMUTJKQqR
zXv+5kKDRvRDlEKtYkkhl2zGyUfizo5OtUy2OXWo8ZfqEtGcob1GqpgOVQ0+us1/
z64cd612/RMkix2B9ETPudMEp8UNHBKz2zRnEsadUquVu+SUnzfUPkTvilSvo4Hx
aSPWSriewi0V9UYKNQXuOvsCAwEAAQ==
`pragma protect end_toolblock

`pragma protect begin_toolblock
`pragma protect key_keyowner = "ipencrypter"
`pragma protect key_keyname = "ipencrypter_key2"
`pragma protect key_method = "rsa"
`pragma protect key_public_key
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA2xEbEbK+w5BWTBoOLQRB
qyK2hYWRC2z85eEncog6pyIisXeoIFCOKfVVCCBm6u+ebKJqmvN8lhmkKICJv7pb
mO/LWJEQwyDsOqJYquTDMC+zqmqc7DuGiuYZ8/XmVpiziDtquPVDztXdnFyFZLMq
wEBNa0zOMWTMZjb2LOzU2jqy04vwuESAV4f1Nkl/96KZUp7pZF1X1jjpNY5UoHAt
20YSn35zdebKdNp1SKAHZDwmxBE+IZmozwmf7hMS2rJElq9UEaN5OeIPUnotQlG6
dDHg7t/5I/XNrtU8WyV2lYR4ZnEbpUtgsRl5OMgFUegGcqpzM1qc5LKBSGI0UdkQ
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
```