

---

RDB OPCUA Server 使用说明  
update: 2022.12

# RDBUASRV

---

支持安全策略和证书的实时库快照 OPCUA SERVER 服务

1 概述.....	1
2 OPCUA 概述.....	1
2.1 安全架构(Security architecture).....	1
2.2 安全策略(SecurityPolicy).....	2
2.3 安全模式(SecurityMode).....	2
2.4 理解证书.....	3
2.5 运行环境.....	4
3 配置和部署.....	5
3.1 配置说明.....	6
3.2 windows 部署.....	7
3.3 Linux 系统部署.....	7
3.3 调试工具.....	8
附录 1 制作自签名服务器证书.....	9

# 1 概述

Rdbuasrv 是实时库自带的一个后台服务程序，将实时库的快照数据通过 OPCUA 协议发布出去，供 OPCUA client 应用程序订阅快照。支持 OPCUA 安全证书，提供 windows/Linux 系统平台部署。

## 2 OPCUA 概述

OPCUA 是 OPC 基金会于 2008 年发布的统一架构跨平台工业信息交换协议，用于替换原 OPCDA 标准，适用于现场设备，控制系统，制造执行系统和企业资源规划系统等应用领域的制造软件。具体参见 OPC 官方网站 (<https://opcfoundation.org/>) 的介绍。

### 2.1 安全架构(Security architecture)

为了更好的理解配置和证书，这里先简单介绍一下 OPCUA 的安全架构。下图为 OPCUA 规范第二章截图。

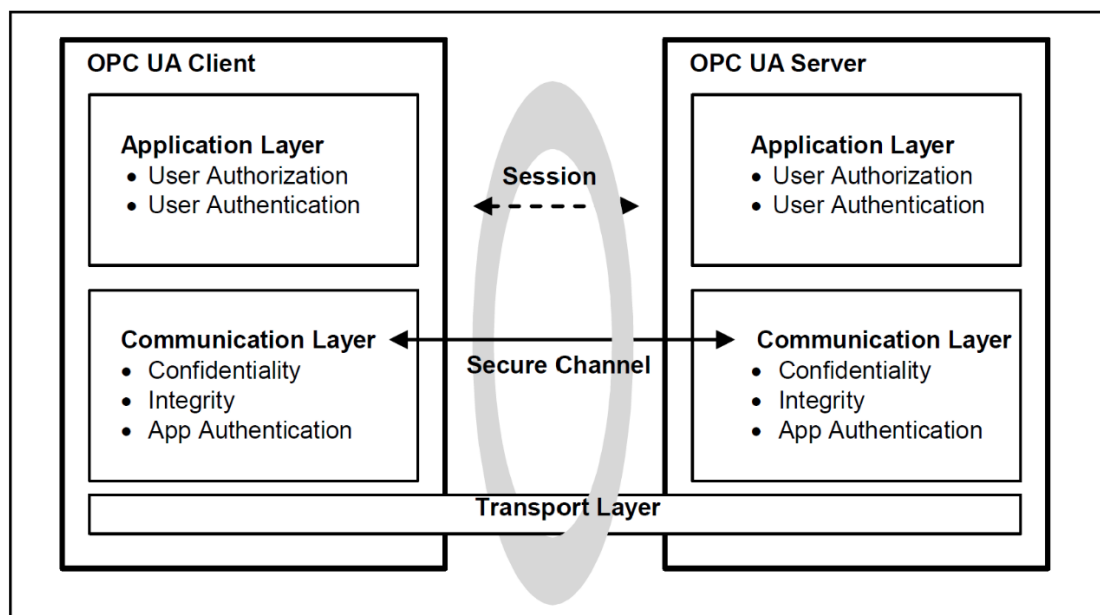


Figure 2 – OPC UA security architecture

图中看出整个通信分为三层：

- 传输层(Transport Layer)：这里一般是 TCP
- 安全通道(Secure Channel)：安全通道，非对称加密握手协商一次性对称加密密钥用于后面的消息加密传输。后面配置的安全策略，证书就是用于这一层的。

- 
- **Session:** 会话层，应用信息编码和交互，包含匿名或者用户/账户模式登录验证。这个和 Secure Channel 没有关系，简单说就是，只要服务端支持，使用带证书的加密通道也可以用匿名方式登录建立会话。

## 2.2 安全策略(SecurityPolicy)

安全策略(SecurityPolicy)定义了使用的加密算法，rdbuasrv 支持的 OPCUA 安全策略包括：

- **None :** <http://opcfoundation.org/UA/SecurityPolicy#None>
- **Basic256:** <http://opcfoundation.org/UA/SecurityPolicy#Basic256>
- **Basic128Rsa15:**  
<http://opcfoundation.org/UA/SecurityPolicy#Basic128Rsa15>
- **Basic256Sha256:**  
<http://opcfoundation.org/UA/SecurityPolicy#Basic256Sha256>

其中 Basic256 和 Basic128Rsa15 已经被标准废弃，不建议使用，已经不安全了。None 是不使用加密和签名的普通通道。

## 2.3 安全模式(SecurityMode)

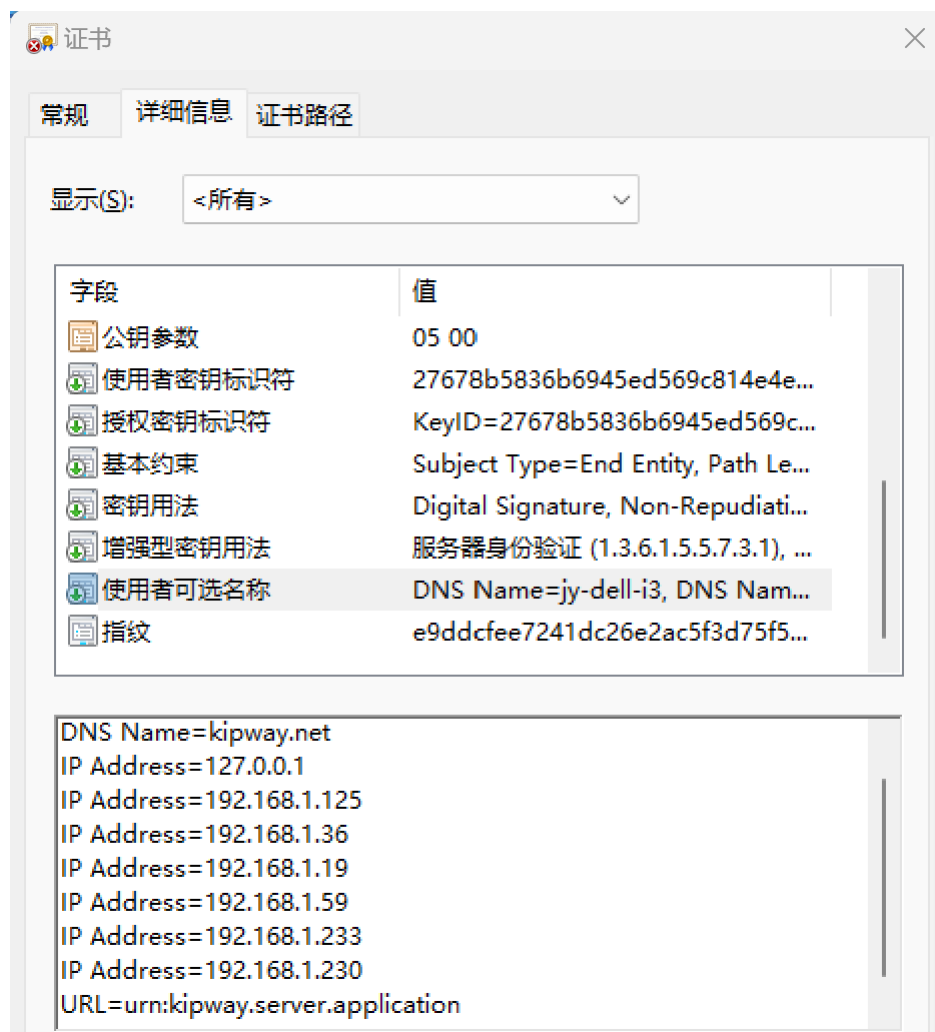
安全模式(SecurityMode)是指对是否对消息做签名或加密。

- **None :** 不签名，也不加密，适合上面的安全策略 None 模式。
- **Sign :** 签名，目的是防止消息被篡改。发送端对消息+双方握手协商出的私密盐参数计算出 SHA256 散列值附在消息尾部，对端也对消息做同样的运算后比对 SHA256 散列值是否相同以判断是否被更改。由于双方握手协商的盐参数第三方无法得知，因此修改消息后无法计算出正确的 SHA256 散列值，很轻易的被接收端识别出消息被篡改。
- **Sign&Encrypt:** 签名和加密一起使用，加密目的是防止被偷窥。发送端使用握手时协商的算法(比如 AES\_CBC)和协商时产生的一次性对称加密密钥(为何不用非对称的 RSA 加密消息，因为这个算法太慢，不适合加密数据，只适合握手协商时传递一次性密钥和 SHA256 散列盐参数使用)。

双方握手过程和 HTTPS 的 TLS 基本相同，TLS 一般不需要客户端证书，只是客户端对服务端的证书的识别采用根证书机构签名识别，OPCUA 需要互相交换证书，握手时发送给对端的信息使用对端的公钥加密，接收端使用私钥解密，防止握手信息被偷窥，支持双方信任模式，可以使用自签名证书。

## 2.4 理解证书

OPCUA 的证书为 X.509 格式的证书，可以用 openssl 工具来制作自签名证书，后面有专门的章节来介绍如何制作证书。如下图是本系统自带证书(rdb\_uaserver\_cert.der)的信息。



证书主要包含应用标识(applicationUri)，非对称加密公钥(一般是 RSA)，域名 IP 等信息，证书在握手时需要发送给对端，如果双方验证后都信任对方，则开始下一步握手协商，否则建立安全通道失败。

与证书配对的还有一个私钥文件，存放的非对称加密（一般是 RSA，2048bit 密钥长度在当前已经够用了）的私钥，要保管好不可泄漏，用来解密对端用本端公钥加密的信息。

RSA 非对称加密的特点是信息使用公钥加密后只能用对应的私钥解密，握手时双方将含有公钥的证书发送给对方后，双方验证并信任对端证书后，交互的握手信息使用对端公钥加密的，保证不会被第三方偷窥到。但是一旦私钥被泄露后，就不安全了，应该把泄露私钥的证书加入 OPCUA Server 的废弃证书列表拒接连接。

OPCUA 支持自签名证书，无需第三方根证书机构签名背书，是否信任对端的证书完全由通信的双方决定。通常大多数的 OPCUA 应用在审核对端证书时有两个证书中的参数必须和

---

环境相同。

DNS 或 IP: 应用程序(本系统 rdbuasrv)所在机器的 DNS 或者 IP 要在“使用者可选名称” DNS 或者 IP 列表中。

应用程序标识(applicationUri)要和“使用者可选名称”中 URL 内容一致。

## 2.5 运行环境

系统环境

Windows:

最低要求: Windows server2008, windows7 即以上系统。推荐 windows server 2012/2016 版。

Linux(X86-64):

内核 3.10 及以上, libstdc++.so.6.0.19, glibc 2.17 及以上; 典型系统 centos7.6; 推荐 centos8 和 ubuntu server1804

Linux(aarch64)

内核 4.4.131 及以上, libstdc++.so.6.0.21, glibc 2.23, 典型系统银河麒麟 V10 系统, 相当于 ubuntu16.04 的 aarch64 系统平台。

无需任何第三方库依赖。

---

## 3 配置和部署

配置文件存放在和 `rdbuasrv/rdbuasrv.exe` 相同目录，文件名 `rdbuasrv.ini`，服务器证书存放在 `cert` 目录下，其余点表 `csv` 文件存放在和 `rdbuasrv` 相同目录。整个配置有三个小结，`[opcua]`，`[rdb]`，`[log]`分别配置 OPCUA 参数，RDB 连接参数，和日志服务器。

```
#rdb opcua server config file

[opcua]

#opcua_port = 4840

#opcua_anonymous_enabled = true

#opcua_username = rdbuasrv

#opcua_userpswd = rdbuasrv


#certificate file in cert/

opcua_cert_file = rdb_uaserver_cert.der


#private key file in cert/

opcua_key_file = rdb_uaserver_key.der


[rdb]

#csv file export from rdb

tagfile = txcloud-gbk.csv


rdburl = ws://127.0.0.1:921

rdbuser = opt1

rdbpswd = opt1


#get snap interval time, millisecond(1/1000 second)

#rdbsnap_interval = 1000
```

---

[log]

logurl = udp://127.0.0.1:999/rdbuasrv?level=dbg

## 3.1 配置说明

注释掉的参数为默认值，如果需要修改，去掉注释，填写参数。

[opcua]小节详解：

**opcua\_port**：服务端口，默认值 **4840**，如果部署的服务器 ip 为 **192.168.1.59**，则 **opcua client** 的连接 url 为 **opc.tcp://192.168.1.59:4840**

**opcua\_anonymous\_enabled**：是否允许匿名连接，默认 **true**

**opcua\_username**：登录账号，默认值 **rdbuasrv**

**opcua\_userpswd**：登录密码，默认值 **rdbuasrv**

**opcua\_cert\_file**：服务器证书文件，默认值空表示不使用证书，这里配置的一个例子证书 **rdb\_uaserver\_cert.der**，只填写文件名，证书存放在 **cert** 子目录中。如果配置了证书，客户端必须支持证书，即使是匿名登录。如果 **opcua** 客户端不支持证书请注释掉这行表示不使用证书和加密。

**opcua\_key\_file**：服务器证书对应的私钥文件，默认值空表示不使用证书。这里配置的一个例子证书 **rdb\_uaserver\_key.der**，只填写文件名，存放在 **cert** 子目录中。注意要和 **opcua\_cert\_file** 一致，**opcua\_cert\_file** 为空，**opcua\_key\_file** 也要为空，否则填写正确配对的 **key** 文件。

**rdbuasrv** 对客户端证书采用信任模式，因此不需要配置信任列表，也不需要配置证书作废和过期信息。

如果 **rdbuasrv** 配置了证书，不支持证书的 **opcua client** 只能使用匿名方式登录。如果 **rdbuasrv** 没有配置证书，不支持证书的 **opcua client** 可以使用用户/账号方式登录。

[rdb]小节

**tagfile**：实时库点表文件，使用 web 版 **dbman** 或者桌面版 **rdbman** 从实时库导出的点表文件。存放在和 **rdbuasrv** 相同目录，只填写文件名。**Tagfile** 中有的标签点才通过 **opcua** 发布，可以自己控制需要配发布的标签。标签表 **csv** 文件可以是 **utf8** 编码，也可以是 **gbk** 编码，但是不要混合编码，否则汉字会出现乱码。

**Rdburl**：实时库连接 url

**rdbuser**：实时库登录账号

**rdbpswd**：实时库登录密码



**rdbsnap\_interval** :快照数据更新间隔, 从实时库中读取快照更新的时间间隔, 默认值是 **1000** 毫秒, 可配置范围为 **200-10000**

[log]小节

**Logurl** : 日志输出服务器的 **url** 和日志级别参数。**Level** 可选 **err,wrn,msg,dbg**

## 3.2 windows 部署

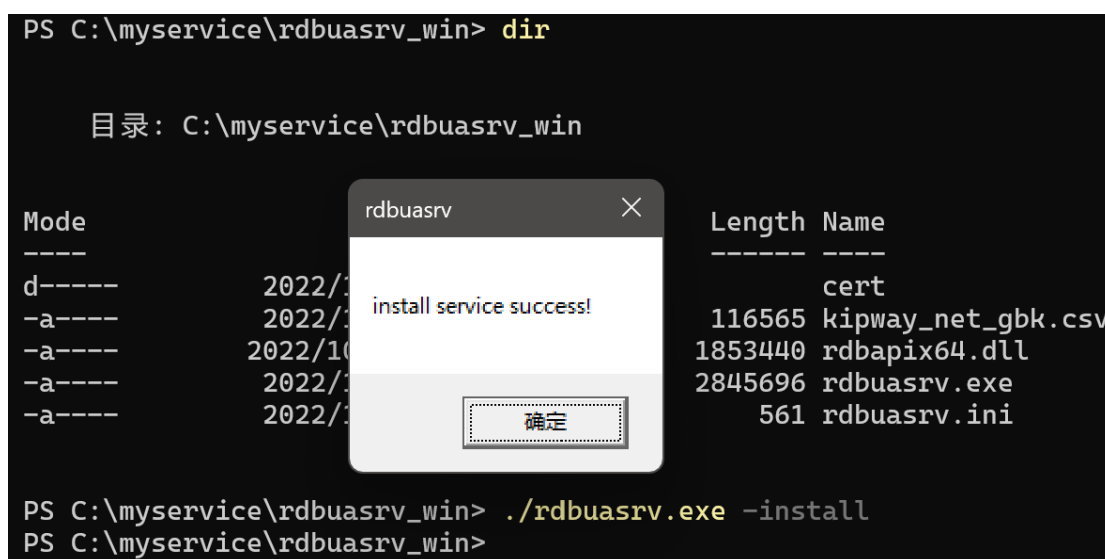
将 **rdbuasrv\_win** 复制带目标系统 **C:/**或者其他目录。先配置好 **rdbuasrv.ini** 文件,然后使用管理员权限的 **cmd** 命令行或者 **powershell** 打开 **rdbuasrv.exe** 所在目录,运行如下命令:

```
PS C:\myservice\rdbuasrv_win> dir

    目录: C:\myservice\rdbuasrv_win

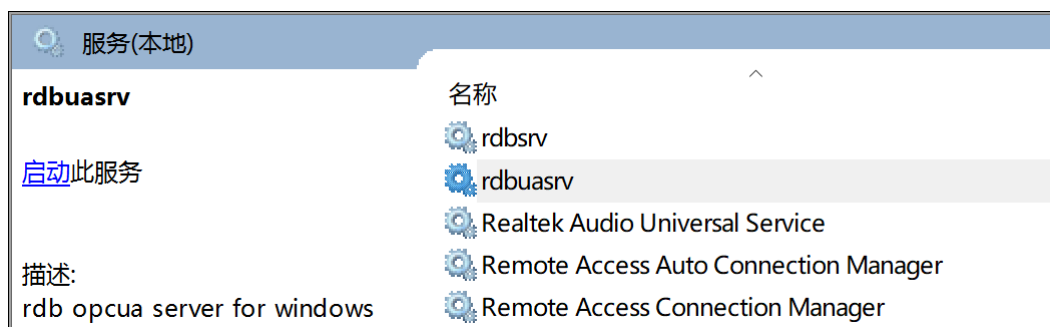
Mode                LastWriteTime         Length Name
----                -
d-----          2022/10/24 10:00             cert
-a----          2022/10/24 10:00        116565 kipway_net_gbk.csv
-a----          2022/10/24 10:00       1853440 rdbapix64.dll
-a----          2022/10/24 10:00       2845696 rdbuasrv.exe
-a----          2022/10/24 10:00         561 rdbuasrv.ini

PS C:\myservice\rdbuasrv_win> ./rdbuasrv.exe -install
PS C:\myservice\rdbuasrv_win>
```



**./rdbuasrv.exe -install**

看到安装成功提示信息后, 到服务里找到 **rdbuasrv** 服务器, 检查配置无误后启动该服务。



## 3.3 Linux 系统部署

将 **rdbuasrv\_linux** 复制到目标系统, 检查配置无误后, 在 **root** 权限下 **cd** 到 **rdbuasrv** 所在目录, 运行如下命令:

---

`./install.sh`

安装脚本会安装到`/usr/local/bin/rdbuasrv`目录下，并创建 `systemd` 服务，并启动 `rdbuasrv` 服务。以后如果要升级可以使用 `systemctl` 来启停 `rdbuasrv` 服务。

```
root@jynuc:/home/rdbuasrv_linux# ls -l
total 6396
drwxr-xr-x 2 root root    4096 Dec  8 09:05 cert
-rwxr-xr-x 1 root root    1125 Dec  8 09:05 install.sh
-rwxr-xr-x 1 root root 116565 Dec  8 09:05 kipway_net_gbk.csv
-rwxr-xr-x 1 root root 2107864 Dec  8 09:05 librdpapix64.so
-rwxr-xr-x 1 root root 3122312 Dec  8 09:05 rdbuasrv
-rwxr-xr-x 1 root root    502 Dec  8 09:07 rdbuasrv.ini
-rwxr-xr-x 1 root root    363 Dec  8 09:05 rdbuasrv.service
-rwxr-xr-x 1 root root 1178629 Dec  8 09:07 txcloud-gbk.csv
root@jynuc:/home/rdbuasrv_linux# vi rdbuasrv.ini
root@jynuc:/home/rdbuasrv_linux# ./install.sh
install as new
'rdbuasrv.service' -> '/etc/systemd/system/rdbuasrv.service'
Created symlink /etc/systemd/system/multi-user.target.wants/rdbuasrv.service -> /etc/s
'rdbuasrv' -> '/usr/local/bin/rdbuasrv/rdbuasrv'
'rdbuasrv.ini' -> '/usr/local/bin/rdbuasrv/rdbuasrv.ini'
'librdpapix64.so' -> '/usr/local/bin/rdbuasrv/librdpapix64.so'
'kipway_net_gbk.csv' -> '/usr/local/bin/rdbuasrv/kipway_net_gbk.csv'
'txcloud-gbk.csv' -> '/usr/local/bin/rdbuasrv/txcloud-gbk.csv'
'cert' -> '/usr/local/bin/rdbuasrv/cert'
'cert/rdb_uaserver_cert.der' -> '/usr/local/bin/rdbuasrv/cert/rdb_uaserver_cert.der'
'cert/rdb_uaserver_key.der' -> '/usr/local/bin/rdbuasrv/cert/rdb_uaserver_key.der'
install success.
start rdbuasrv

rdbuasrv is runing!
```

启动：`sudo systemctl start rdbuasrv`

停止：`sudo systemctl stop rdbuasrv`

工作目录在`/usr/local/bin/rdbuasrv`，配置文件和点表存放在该目录，制作的自签名服务器证书存放在`/usr/local/bin/rdbuasrv/cert`子目录下。

更改 `rdbuasrv.ini` 或者点表文件内容需要重新启动 `rdbuasrv` 服务才生效。

Linux aarch64 平台的安装方式相同，安装文件在 `rdbuasrv_aarch64` 目录。

## 3.3 调试工具

本系统在以下第三方 UA client 工具下测试通过：

UaExpert v1.6.3-448

Prosys OPC UA Client v2.100.3-186

在本系统 `ioserver` 自带的 `dac_opcux` 驱动下测试通过。

---

## 附录 1 制作自签名服务器证书

如果条件满足（主要是 IP 地址），可以使用 `rdbuasrv` 自带的有效期长达 10 年的服务器证书 `rdb_uaserver_cert.der`。

可按照如下方法自己制作证书。

本驱动附带了一个 `python3` 脚本，从 `open62541` 项目修改而来，去掉了 `python3` 组件 `netifaces` 的依赖，方便使用。

选择 `ubuntu20.04` 或者 `22.04` 桌面版系统，开发工作的最佳 Linux 平台，默认安装常用工具比如 `perl`, `openssl`, `python3` 等，方便直接使用。

将 `uacert` 目录复制到目标 `ubuntu` 系统。

使用 `vim` 修改 `python3` 脚本 `create_self-signed.py` 文件

修改其中的 DNS2 和 IP 列表, 如下图。

```
#set IP address
os.environ['IPADDRESS1'] = "127.0.0.1"
os.environ['IPADDRESS2'] = "192.168.1.125"
os.environ['IPADDRESS3'] = "192.168.1.36"
os.environ['IPADDRESS4'] = "192.168.1.19"
os.environ['IPADDRESS5'] = "192.168.1.59"
os.environ['IPADDRESS6'] = "192.168.1.233"
os.environ['IPADDRESS7'] = "192.168.1.230"

os.environ['HOSTNAME'] = socket.gethostname()

#填写其他的DNS2
os.environ['HOSTNAME2'] = "kipway.net"
openssl_conf = os.path.join(certsdire, "localhost.cnf")

os.chdir(os.path.abspath(args.outdir))
```

然后执行

```
./create_self-signed.py -u urn:kipway.server.application -c myuaserver
```

其中 `myuaserver` 是证书名不需要加扩展名，改成自己喜欢的名字。

证书测试，推荐使用 `Prosys OPC UA Client` 作为 OPCUA client 端工具。