TÀI LIỆU HƯỚNG DẪN TÍCH HỢP BỘ CÔNG CỤ KÝ SỐ THEO NGHỊ ĐỊNH 30/2020/NĐ-CP

NỘI DUNG

1 Về tài liệu hướng dẫn tích hợp tích hợp 4

1.1 Mục đích 4

1.2 Đối tượng sử dụng 4

1.3 Về công cụ tích hợp 4

2 Quy trình tích hợp 5

3 Xây dựng ứng dụng web hỗ trợ upload file lên hệ thống 5

4 Mô tả API ký số của thư viện vgcaplugin.js 7

4.1 Hàm vgca_sign_msg 7

4.2 Hàm vgca_sign_issued 8

4.3 Hàm vgca_sign_ approved 9

4.4 Hàm vgca_comment 9

4.5 Hàm vgca_sign_income 10

4.6 Hàm vgca_sign_appendix 10

4.7 Hàm vgca_sign_copy 11

4.8 Hàm vgca_sign_files 12

4.9 Hàm vgca_sign_xml 12

4.10 Hàm vgca_verify_xml 13

4.11 Hàm vgca_sign_json 14

4.12 Hàm vgca_verify_json 14

5 Hướng dẫn khai báo hàm gọi API ký số 15

5.1 Hàm xử lý kết quả SignFileCallBack 15

5.2 Hàm gọi API ký số 16

6 Hướng dẫn cài đặt, cấu hình tool VGCASignService 17

6.1 Yêu cầu đối với hệ thống sử dụng tool 17

6.2 Hướng dẫn checksum file cài đặt 17

6.3 Hướng dẫn kiểm tra chữ ký số file cài đặt 18

6.4 Cài đặt tool VGCASignService 21

6.5 Hướng dẫn cấu hình dịch vụ chứng thực 23
6.6 Hướng dẫn cấu hình mẫu chữ ký 25
7 Thông tin liên hệ hỗ trợ.....29
7.1 Cục Chứng thực số và Bảo mật thông tin 29
7.2 Bộ phận Hỗ trợ kỹ thuật ..... **Error! Bookmark not defined.**
7.3 Bộ phận Nghiên cứu ứng dụng ..... **Error! Bookmark not defined.**

1 Về tài liệu hướng dẫn tích hợp tích hợp

1.1 Mục đích

Tài liệu hướng dẫn tích hợp này mô tả chi tiết kỹ thuật và các nội dung cần chú ý trong quá trình các cơ quan, đơn vị sử dụng, tích hợp dịch vụ chứng thực chữ ký số do Ban Cơ yếu Chính phủ cung cấp vào các hệ thống thông tin phát triển trên nền tảng .Net Framework trong đó bao gồm:

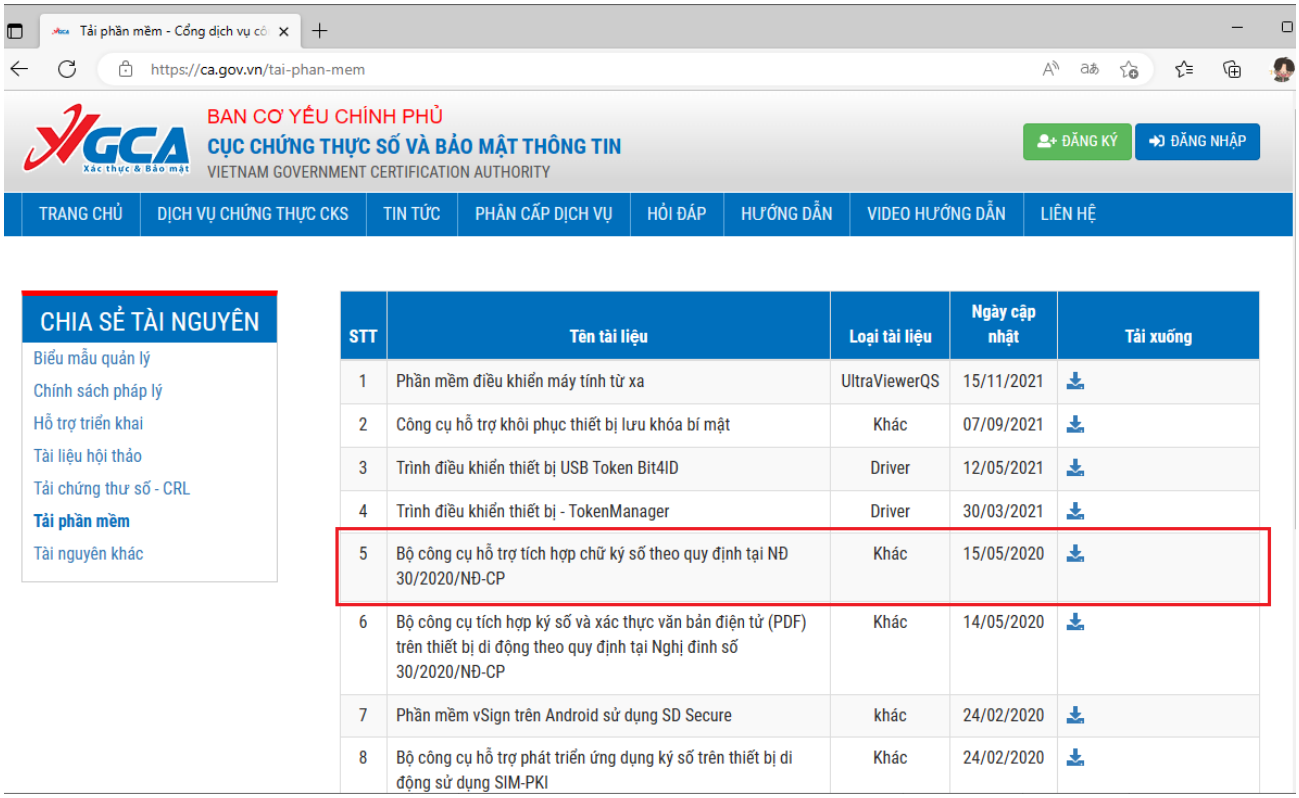
- Quy trình thực hiện ích hợp.
- Mô tả chi tiết API ,trong thư viện vgcaplugin.js cung cấp .
- Hướng dẫn cấu hình tool ký số VGCA SignService.

1.2 Đối tượng sử dụng

Cán bộ kỹ thuật, cán bộ phát triển phần mềm, ứng dụng của các cơ quan bộ, ngành, địa phương triển khai tích hợp trên các hệ thống thông tin.

1.3 Về công cụ tích hợp

Bộ công cụ ký số theo Nghị định 30/2020/NĐ-CP do Ban Cơ yếu Chính phủ cung cấp được publish trên trang chủ của Cục Chứng thực số và Bảo mật thông tin. Để tải về bộ công cụ tích hợp, truy cập vào trang chủ của Cục Chứng thực số và Bảo mật thông tin theo địa chỉ https://dichvucong.ca.gov.vn/page/taitainguyen/PM :



Bộ công cụ bao gồm:

TT	Tên thư mục/file	Mô tả
1	VGCASignServiceSetup.msi	Tool ký số với USB Token cài đặt trên máy tính người dùng.
	Mã băm SHA1	6e413ccbafc85546218068d4718b263ebe8ed9cb
	Mã băm SHA256	669f6464181c07e7446c1f80ae494e1be0ec376976b98a4e3a8161023c02d6c7
2	doc	Thư mục lưu tài liệu hướng dẫn tích hợp và hướng dẫn sử dụng bộ công cụ
3	js	Thư mục lưu thư viện nhúng hỗ trợ tích hợp
4	Sample	Thư mục lưu mã nguồn mẫu hướng dẫn tích hợp

2 Quy trình tích hợp

Để triển khai tích hợp và sử dụng công cụ ký số tài liệu, thực hiện các bước như sau:

Bước 1: Cài đặt tool VGCASignServiceSetup.msi trên máy tính người dùng để thực hiện ký số (Xem hướng dẫn chi tiết ở mục 4)

Bước 2: Xây dựng ứng dụng web hỗ trợ upload file lên hệ thống: FileUploadHandler.aspx.

Bước 3: Nhúng thư viện JavaScript vgcaplugin.js lên site để người dùng tương tác gọi chức năng ký số

```
<script type="text/javascript" src="./vgcaplugin.js"></script>
```

Bước 4. Khai báo các hàm Javascript gọi API ký số và lấy phiên bản tool ký số trên site.

3 Xây dựng ứng dụng web hỗ trợ upload file lên hệ thống

Bước 1: Tạo ứng dụng web FileUploadHandler.aspx

Bước 2: Khai báo hàm Upload() thực hiện tải file lên hệ thống. Cấu trúc hàm Upload():

```
private void Upload()
{
```

```
try
{
    var builder = new UriBuilder(Request.Url.Scheme, Request.Url.Host, Request.Url.Port);
    HttpPostedFile file = base.Request.Files["uploadfile"];
    string path = Path.GetFileName(file.FileName);
    string fileExt = Path.GetExtension(path).ToLower();
    string uploadFilename = string.Format("{0}.signed{1}", Path.GetFileNameWithoutExtension(path), fileExt);
    string str = string.Format("{0}Upload/{1}", builder.ToString(), uploadFilename);
    file.SaveAs(base.Server.MapPath("~/Upload/" + uploadFilename));
    this.Page.Response.Write("{\"Status\":true, \"Message\": \"\", \"FileName\": \"" + path + "\", \"FileServer\": \"" + str + "\"}");
}
catch (Exception ex)
{
    this.Page.Response.Write("{\"Status\":false, \"Message\": \"" + ex.Message + "\", \"FileName\": \"\", \"FileServer\": \"\"}");
}
    }
```

Kết quả: Ghi chuỗi có định dạng JSON chứa kết quả upload file vào HTTP Response. Chuỗi kết quả có cấu trúc:

```
{
    Status: trạng thái upload file (True/False),
    Message: Thông báo lỗi nếu upload không thành công (string),
    FileName: tên file upload (string),
    FileServer: đường dẫn lưu file upload trên server (string)
}
```

Bước 3: Xử lý quá trình web được load lên

```
protected void Page_Load(object sender, EventArgs e)
    {
        this.Page.Response.Clear();
        if (base.Request.Files["uploadfile"] != null)
        {
            Upload();

        }
        else if (base.Request.QueryString["download"] != null)
        {
            string filename = base.Server.MapPath("~/congvan.pdf");
            System.Web.HttpResponse response = System.Web.HttpContext.Current.Response;
            response.ClearContent();
            response.Clear();
            response.ContentType = "application/pdf";
            response.AddHeader("Content-Disposition", "attachment; filename=" + "testfilename.pdf" + ";");
```

```
            response.TransmitFile(filename);
            response.Flush();
        }
        this.Page.Response.End();
    }
```

Bước 4: Publish ứng dụng web vừa tạo lên server

4 Mô tả API ký số của thư viện vgcaplugin.js

4.1 Hàm vgca_sign_msg

- Mục đích sử dụng: Dùng để ký số nội dung dữ liệu báo cáo được đóng gói theo định dạng JSON đã được quy đinh tại Quyết định số 2337/QĐ-BTTTT ngày 31/12/2019 của Bộ Thông tin và Truyền thông.
- Giao diện hàm: `vgca_sign_msg(sender, prms, CallBackFunc)`

  + sender: Tham số truyền thông tin cho quá trình xử lý kết quả, ví dụ: ID Button Ký số
  + prms: Tham số ký số
  + CallbackFunc: Hàm callback xử lý kết quả trả về

- Giao diện hàm Callback: `CallbackFunc(sender, evData)`

  + sender: Thông tin được truyền từ hàm thực hiện ký số vgca_sign_msg (Ví dụ: ID Button ký số, khi nhận được chữ ký số có thể kích hoạt event click để push dữ liệu lên Server)
  + evData: Kết quả trả về

- Cấu trúc tham số đầu vào prms:

```
{
    "Encode": chuẩn đóng gói dữ liệu (RSA/ECDSA/PKCS#7)
    "HashValue": "Chuỗi Base64 giá trị băm",
    "HashAlg": "SHA256" //Thuật toán băm
}
```

- Cấu trúc thông tin trả về evData:

```
{
    "Status": 0, //Mã lỗi: 0-Thành công, x- lỗi
    "Message": "….", //Mô tả lỗi
    "Signature": "Base64 chữ ký đóng gói theo RSA/ECDSA/PKCS#7 Detached",
    "CertBase64": "chuỗi base64 của chứng thư số ký số"
}
```

4.2 Hàm vgca_sign_issued

- Mục đích: Gọi vgcaplugin, hiển thị cửa sổ giao diện ký số (đóng dấu) phát hành văn bản theo quy định tại Nghị định 30/2020.

- Giao diện hàm: `vgca_sign_issued(prms, SignFileCallBack)`

  + `prms: Tham số ký số`

  + `SignFileCallBack: Hàm callback xử lý kết quả trả về`

- Cấu trúc tham số đầu vào prms :

```
{
    "FileUploadHandler": //url của ứng dụng web upload file FileUploadHandler.aspx đã tạo,
    "SessionId": ////phiên làm việc của người đang truy cập, trường hợp hệ thống thông tin yêu cầu đăng nhập sử dụng
    "JWTToken": //nếu sử dụng jwt token để xác thực người dùng, trường hợp hệ thống thông tin yêu cầu đăng nhập sử dụng
    "FileName": //đường dẫn văn bản ký số
    "DocNumber": //Cấp số cho văn bản (công văn) phát hành
    "IssuedDate": //Ngày ký, đóng dấu phát hành văn bản
}
```

- Giao diện hàm Callback: `SignFileCallBack(sender, rv)`

+ sender: Thông tin được truyền từ hàm thực hiện ký số vgca_sign_xml (Ví dụ: ID Button ký số, khi nhận được chữ ký số có thể kích hoạt event click để push dữ liệu lên Server)

+ rv: Kết quả trả về

```
{
    DocumentDate:" "//Ngày văn bản
    DocumentNumber: ""//Số văn bản
    FileName: ""//Đường dẫn văn bản cần ký số
    FileServer: ""//Đường dẫn văn bản đã ký số
    Message: ""//Mô tả lỗi
    Status: ////Mã lỗi: 0-Thành công, x- lỗi
}
```

4.3 Hàm vgca_sign_ approved

- Mục đích: Gọi vgcaplugin, hiển thị cửa sổ giao diện ký số phê duyệt văn bản theo quy định tại Nghị định 30/2020.
- Giao diện hàm: `vgca_sign_approved(prms,SignFileCallBack)`
- Cấu trúc tham số đầu vào prms :

```
{
    "FileUploadHandler": //url của ứng dụng web upload file FileUploadHandler.aspx đã tạo,
    "SessionId": //phiên làm việc của người đang truy cập, trường hợp hê thống thông tin yêu cầu đăng nhập sử dụng
    "JWTToken": //nếu sử dụng jwt token để xác thực người dùng, trường hợp hệ thống thông tin yêu cầu đăng nhập sử dụng
    "FileName": //đường dẫn văn bản ký số
}
```

4.4 Hàm vgca_comment

- Mục đích: Gọi vgcaplugin, hiển thị cửa sổ giao diện thêm ý kiến, chỉ đạo cho văn bản theo quy định tại Nghị định 30/2020.
- Giao diện hàm: `vgca_ comment (prms, SignFileCallBack)`
- Cấu trúc tham số đầu vào prms :

```
{
    "FileUploadHandler": //url của ứng dụng web upload file FileUploadHandler.aspx đã tạo,
    "SessionId": //phiên làm việc của người đang truy cập, trường hợp hê thống thông tin yêu cầu đăng nhập sử dụng
    "JWTToken": //nếu sử dụng jwt token để xác thực người dùng, trường hợp hệ thống thông tin yêu cầu đăng nhập sử dụng
    "FileName": //đường dẫn văn bản ký số
    Metadata": //dữ liệu phản hồi, bổ sung tùy chọn
}
```

- Cấu trúc tham số Metadata:

```
{
    "Key": string,
    "Value": string
}
```

4.5 Hàm vgca_sign_income

- Mục đích: Gọi vgcaplugin, hiển thị cửa sổ giao diện ký số văn bản đến theo quy định tại Nghị định 30/2020.
- Giao diện hàm `vgca_sign_income(prms, SignFileCallBack)`
- Cấu trúc tham số đầu vào prms :

```
{
    "FileUploadHandler": //url của ứng dụng web upload file FileUploadHandler.aspx đã tạo,
    "SessionId": //phiên làm việc của người đang truy cập, trường hợp hê thống thông tin yêu cầu đăng nhập sử dụng
    "JWTToken": //nếu sử dụng jwt token để xác thực người dùng, trường hợp hệ thống thông tin yêu cầu đăng nhập sử dụng
    "FileName": //đường dẫn văn bản đến
    "Metadata": //dữ liệu phản hồi, bổ sung tùy chọn
}
```

- Định dạng Metadata:

```
{
    "Key": string,
    "Value": string
}
```

4.6 Hàm vgca_sign_appendix

- Mục đích: Gọi vgcaplugin, hiển thị cửa sổ giao diện ký số văn bản đính kèm theo quy định tại Nghị định 30/2020.
- Giao diện hàm: `vgca_sign_appendix(prms, SignFileCallBack)`
- Cấu trúc tham số đầu vào prms :

```
{
    "FileUploadHandler": //url của ứng dụng web upload file FileUploadHandler.aspx đã tạo,
    "SessionId": //phiên làm việc của người đang truy cập, trường hợp hê thống thông tin yêu cầu đăng nhập sử dụng
    "JWTToken": //nếu sử dụng jwt token để xác thực người dùng, trường hợp hệ thống thông tin yêu cầu đăng nhập sử dụng
    "FileName": //đường dẫn tài liệu đính kèm
```

```
    "DocNumber": //số công văn của văn bản cần đính kèm tài liệu
    "Metadata": //dữ liệu phản hồi, bổ sung tùy chọn
}
```

- Định dạng Metadata:

```
{
    "Key": string,
    "Value": string
}
```

4.7 Hàm vgca_sign_copy

- Mục đích: Gọi vgcaplugin, hiển thị cửa sổ giao diện ký số bản sao điện tử (sao y, sao lục, trích sao)của văn bản gốc theo quy định tại Nghị định 30/2020.
- Giao diện hàm: `vgca_sign_copy(prms, SignFileCallBack)`
- Cấu trúc tham số đầu vào prms :

```
{
  "FileUploadHandler": //url của ứng dụng web upload file FileUploadHandler.aspx đã tạo,
    "SessionId": //phiên làm việc của người đang truy cập, trường hợp hê thống thông tin yêu cầu đăng nhập sử dụng
    "JWTToken": //nếu sử dụng jwt token để xác thực người dùng, trường hợp hệ thống thông tin yêu cầu đăng nhập sử dụng
  "FileName": //đường dẫn tài liệu đính kèm
  "DocNumber"://số công văn của văn bản gốc cần tạo bản sao điện tử
  "Metadata": //dữ liệu phản hồi, bổ sung tùy chọn
}
```

- Định dạng Metadata:

```
{
    "Key": string,
    "Value": string
}
```

4.8 Hàm vgca_sign_files

- Mục đích: Gọi vgcaplugin, hiển thị cửa sổ giao diện ký nhiều tệp văn bản theo mẫu số hóa, văn bản đến.

- Giao diện hàm:`vgca_sign_files(json_prms, SignFileCallBack)`

- Cấu trúc tham số đầu vào prms định dạng json, có cấu trúc:

```
{
 "FileUploadHandler": "",//url của ứng dụng web upload file FileUploadHandler.aspx đã tạo
 "SessionId": //phiên làm việc của người đang truy cập, trường hợp hệ thống thông tin yêu cầu đăng nhập sử dụng
 "JWTToken": //nếu sử dụng jwt token để xác thực người dùng, trường hợp hệ thống thông tin yêu cầu đăng nhập sử dụng
 "Files": [//Danh sách files cần ký số
     {
         "FileID": "",//ID của tệp ký số
         "FileName": "",//tên tệp
         "URL": ""//Đường dẫn tệp ký số
     },
     …
     ]
 }
```

4.9 Hàm vgca_sign_xml

- Mục đích: Gọi vgcaplugin, hiển thị cửa sổ giao diện ký số dữ liệu XML.

- Giao diện hàm: `vgca_sign_xml(sender,json_prms,SignXMLCallBack)`:

+ sender: Tham số truyền thông tin cho quá trình xử lý kết quả, ví dụ: ID Button Ký số

+ json_prms: Tham số ký số

+ SignXMLCallBack: Hàm callback xử lý kết quả trả về

- Giao diện hàm Callback: `SignXMLCallBack(sender,rv)`

+ sender: Thông tin được truyền từ hàm thực hiện ký số vgca_sign_xml (Ví dụ: ID Button ký số, khi nhận được chữ ký số có thể kích hoạt event click để push dữ liệu lên Server)

+ rv: Kết quả trả về

- Cấu trúc tham số đầu vào json_prms:

```
{
   "Base64Content": "",// Chuỗi Base64 giá trị băm của dữ liệu xml
   "ReturnSignatureOnly": "true" or "false",//dạng chữ ký số trả về
   "HashAlg": “”,//Thuật toàn hàm băm
   "XmlDsigForm": "true" or "false"//chuẩn
}
```

- Cấu trúc thông tin trả về rv:

```
{
    “Status”: 0, //Mã lỗi: 0-Thành công, x- lỗi
    “Message”: “….”, //Mô tả lỗi
    “Signature”: “Base64 chữ ký đóng gói theo PKCS#7 Detached”
}
```

4.10 Hàm vgca_verify_xml

- Mục đích: Gọi vgcaplugin, hiển thị giao diện ký nhiều tệp văn bản theo mẫu số hóa, văn bản đến.

- Giao diện hàm:`vgca_verify_xml(json_prms,VerifyXMLCallBack)`

- Cấu trúc tham số đầu vào json_prms định dạng json, có cấu trúc :

```
{
“Data”: //dữ liệu xác thực
“Format”:// định dạng dữ liệu xác thực
}
```

- VerifyXMLCallBack: Hàm callback xử lý kết quả trả về. Giao diện hàm: `VerifyXMLCallBack(rv)` ,rv: Kết quả trả về

- Cấu trúc thông tin trả về rv:

```
    {
  “Status”: 0, //Mã lỗi: 0-Thành công, x- lỗi
  “Message”: “”, //Mô tả lỗi
  “Signature”: “” //thông tin xác thực
    }
```

4.11 Hàm vgca_sign_json

- Mục đích: Gọi vgcaplugin, hiển thị cửa sổ giao diện ký số dữ liệu XML.

- Giao diện hàm: `vgca_sign_json(sender,json_prms,SignJSONCallBack)`

+ sender: Tham số truyền thông tin cho quá trình xử lý kết quả, ví dụ: ID Button Ký số

+ json_prms: Tham số ký số

+ SignJSONCallBack: Hàm callback xử lý kết quả trả về

- Giao diện hàm Callback: `SignJSONCallBack(sender,rv)`

+ sender: Thông tin được truyền từ hàm thực hiện ký số vgca_json_xml (Ví dụ: ID Button ký số, khi nhận được chữ ký số có thể kích hoạt event click để push dữ liệu lên Server)

+ rv: Kết quả trả về

- Cấu trúc tham số đầu vào json_prms:

```
{
   "JsonContent": "",// Chuỗi Base64 giá trị băm của dữ liệu xml
   "DetachPayload": "true" or "false",//
   "IsPss": "true" or "false",//lược đồ chữ ký số
}
```

- Cấu trúc thông tin trả về rv:

```
{
    "Status": 0, //Mã lỗi: 0-Thành công, x- lỗi
    "Message": "….", //Mô tả lỗi
    "Signature": "Base64 chữ ký đóng gói theo PKCS#7 Detached"
}
```

4.12 Hàm vgca_verify_json

- Mục đích: Gọi vgcaplugin, hiển thị giao diện ký nhiều tệp văn bản theo mẫu số hóa, văn bản đến.

- Giao diện hàm:`vgca_verify_json(content, VerifyJSONCallBack)`

- Cấu trúc tham số đầu vào json_prms định dạng json, có cấu trúc :

```
{
"content": //dữ liệu xác thực
```

}

- VerifyJSONCallBack: Hàm callback xử lý kết quả trả về. Giao diện hàm: `VerifyXMLCallBack(rv)` ,rv: Kết quả trả về

- Cấu trúc thông tin trả về rv:

```
{
    "Status": 0, //Mã lỗi: 0-Thành công, x- lỗi
    "Message": "", //Mô tả lỗi
    "JsonContent": "", //Dữ liệu json decode
}
```

5 Hướng dẫn khai báo hàm gọi API ký số

5.1 Hàm xử lý kết quả SignFileCallBack

- Mục đích: Hàm callback xử ký kết quả trả về, hiển thị thông tin ký số trên văn bản trả về
- Giao diện hàm: `SignFileCallBack(evData)`
- Cấu trúc tham số đầu vào evData:

```
{
    "Status": 0, //Mã lỗi: 0-Thành công, x- lỗi
    "Message": "….", //Mô tả lỗi
    "DocumentNumber": "….", //Mô tả số của văn bản (công văn)
    "DocumentDate": "….", //Mô tả ngày ký số
    "FileName": "….", //Mô tả tên văn bản
    "FileServe": "….", //Mô tả đường dẫn văn bản trên server
}
```

- Khai báo hàm xử lý

```
function SignFileCallBack(evData) {
        var received_msg = JSON.parse(evData);
        console.log(received_msg);
        if (received_msg.Status == 0) {
          //console.log(received_msg);
          var fileName = received_msg.FileName;//lấy tên văn bản ký
          var url = received_msg.FileServer;//lấy link lưu văn bản trên server
          var docNum = received_msg.DocumentNumber;//lấy số công văn ký số trên văn
bản
          var docDate = received_msg.DocumentDate;//lấy ngày ký số văn bản
          } else {
```

```
                var error = received_msg.Message;//lỗi ký số
            }
        }
```

5.2 Hàm gọi API ký số

Hàm javascript gọi API "vgca_sign_ approved" ký phê duyệt văn bản

```
<script type="text/javascript">
    function exc_sign_approved(url) {
        var prms = {}; //Tham số ký
        //FileUploadHandlerLink - đến ứng dụng web đã tạo (mục 2.1) hỗ trợ upload file lên server
        prms["FileUploadHandler"] = ".../FileUploadHandler.aspx";
        //SessionId - Phiên làm việc của người đang truy cập
        prms["SessionId"] = "";
        //FileName - Đường dẫn đến văn bản cần ký số được lưu trên server.
        prms["FileName"] = ""; //Trường hợp văn bản cần ký lưu local, FileName = "";
        var json_prms = JSON.stringify(prms);
        //Gọi API ký số
        vgca_sign_approved(prms, SignFileCallBack);
        }
</script>
```

Hàm gọi API "vgca_sign_issued" ký đóng dấu phát hành văn bản

```
<script type="text/javascript">
    function exc_sign_issued() {
        var prms = {}; //Tham số ký
        //FileUploadHandlerLink - đến ứng dụng web đã tạo (mục 2.1) hỗ trợ upload file lên server
        prms["FileUploadHandler"] = ".../FileUploadHandler.aspx";
        //SessionId - Phiên làm việc của người đang truy cập
        prms["SessionId"] = "";
        //FileName - Đường dẫn đến văn bản cần ký số được lưu trên server.
        prms["FileName"] = ""; //Trường hợp văn bản cần ký lưu local, FileName = "";
        prms["DocNumber"] = "";//Cấp số công văn cho văn bản, có thể cấp trên giao diện của tool ký số
        prms["IssuedDate"] = "";//Ngày ký đóng dấu phát hành văn bản
        var json_prms = JSON.stringify(prms);
    //Gọi API ký số
        vgca_sign_issued(json_prms, SignFileCallBack1);
    }
</script>
```

Thực hiện tương tự với các API còn lại.

6 Hướng dẫn cài đặt, cấu hình tool VGCASignService

6.1 Yêu cầu đối với hệ thống sử dụng tool

- Hệ điều hành: Tool sử dụng cho các hệ điều hành Windows phiên bản XP SP3 trở lên.

- Bộ nhớ RAM: 512Mb trở lên.

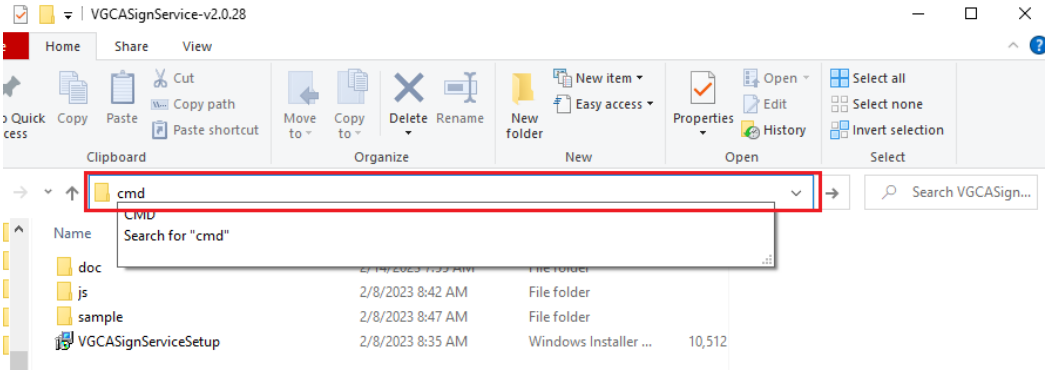
- Dung lượng ổ đĩa: 10Gb trở lên.

- Trên máy tính người dùng cần cài đặt .Net Framework 4.0.

6.2 Hướng dẫn checksum file cài đặt

Mục đích: Kiểm tra tính toàn vẹn của file cài đặt, xác định file cài đặt chưa bị thay đổi.

Bước 1: Mở thư mục lưu file cài đặt tool VGCASignService. Gõ "cmd" trên thanh địa và nhấn **Enter**.



Bước 2. Trên giao diện Command Line CMD của Windows nhập lệnh:

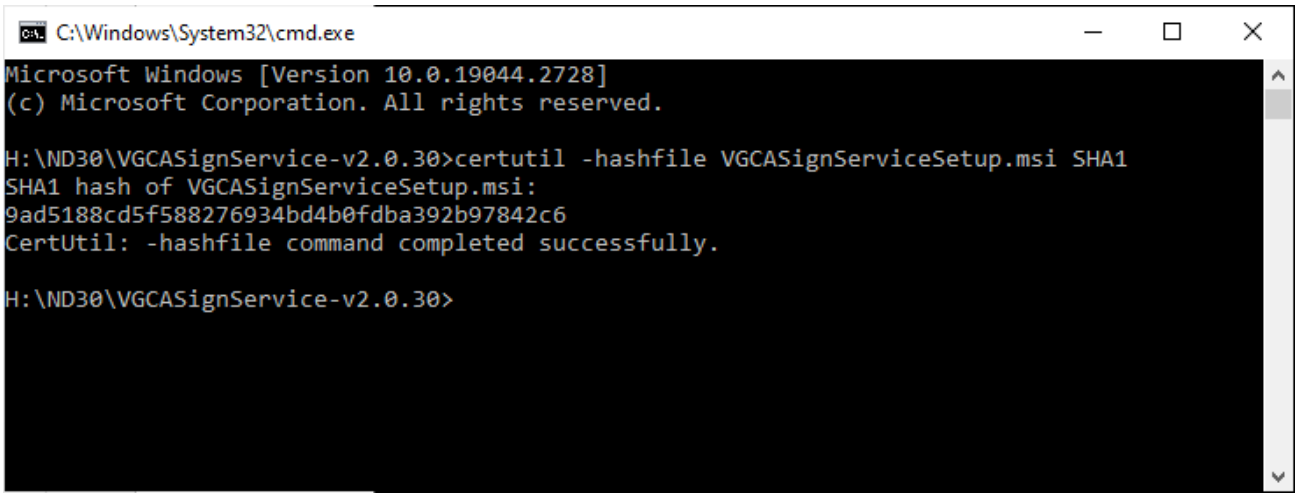
```
Certutil -hashfile <file> Loai_Hash
```

Trong đó:

`<file>`: Tên file, đường dẫn file cần checksum trên máy tính

`<Loai_Hash>`: Hàm băm cần kiểm tra. Cần kiểm tra hai hàm băm: SHA1 và SHA256, .

Ví dụ:



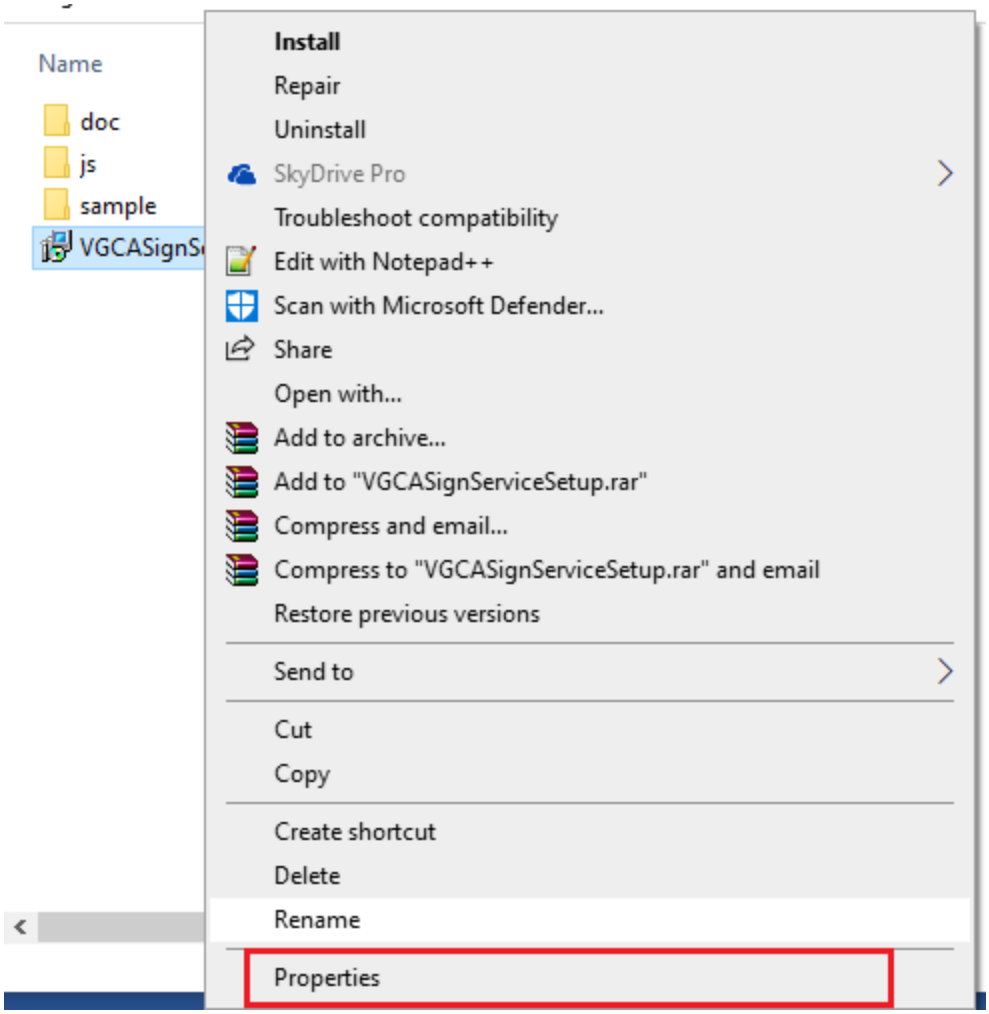
Bước 3. So sách mã băm nhận được với mã băm của hàm băm tương ứng được cung cấp.

- Nếu hai mã băm giống nhau, file cài đặt chưa bị thay đổi => tiến hành cài đặt công cụ để sử dụng.

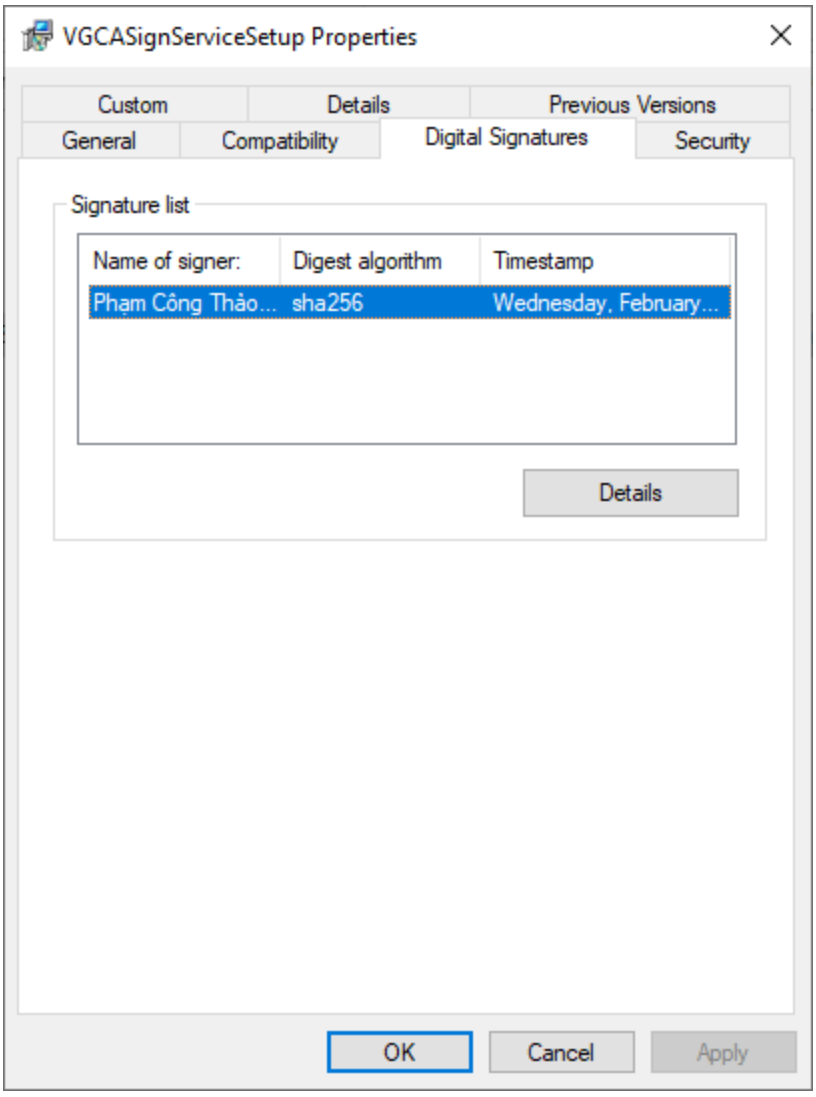
- Nếu hai mã băm khác nhau, file cài đặt đã bị thay đổi: Dừng cài đặt công cụ, thông báo với cơ quan cung cấp Bộ công cụ để kiểm tra.

6.3 Hướng dẫn kiểm tra chữ ký số file cài đặt

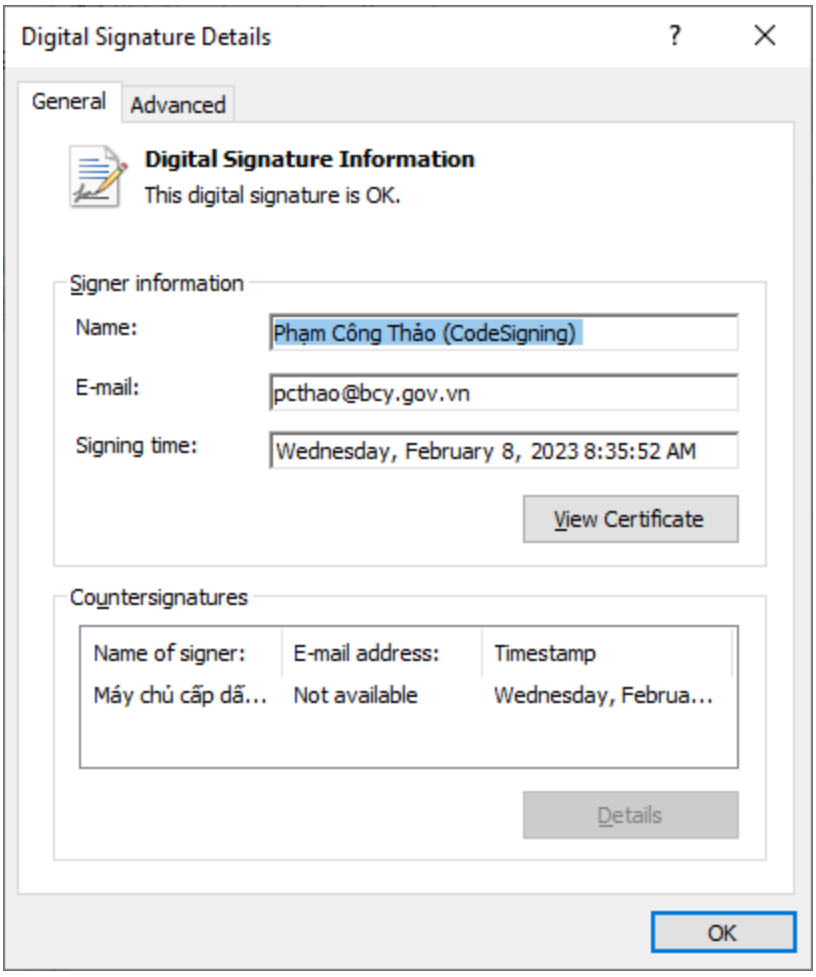
Bước 1. Click chuột phải ở file cài đặt, chọn "**Properties**"



Bước 2. Chọn mục "**Digital Signature**" vào chọn chữ ký số cần kiểm tra và nhấn "Detail" để hiển thị kết quả kiểm tra chữ ký số

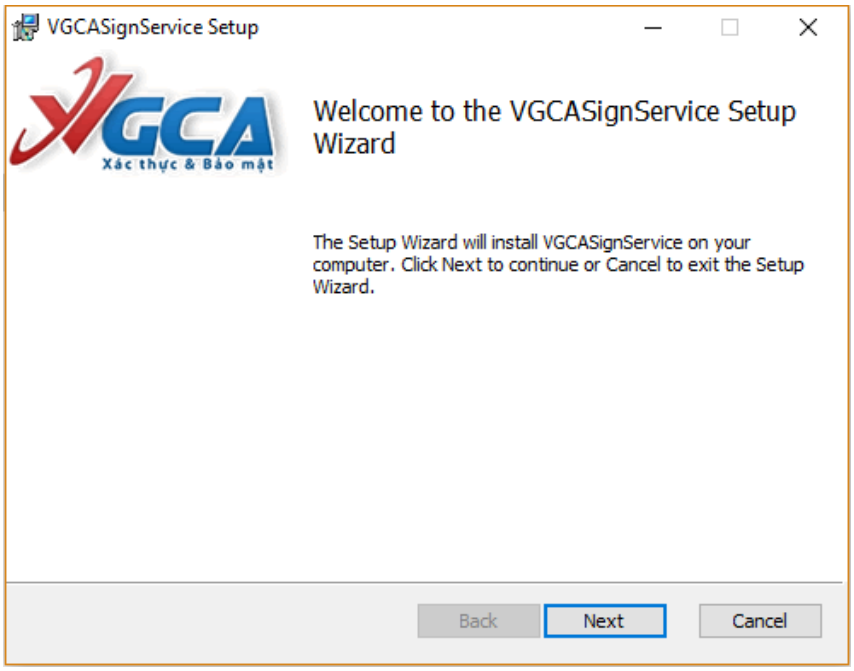


Kết quả kiểm tra chữ ký số:

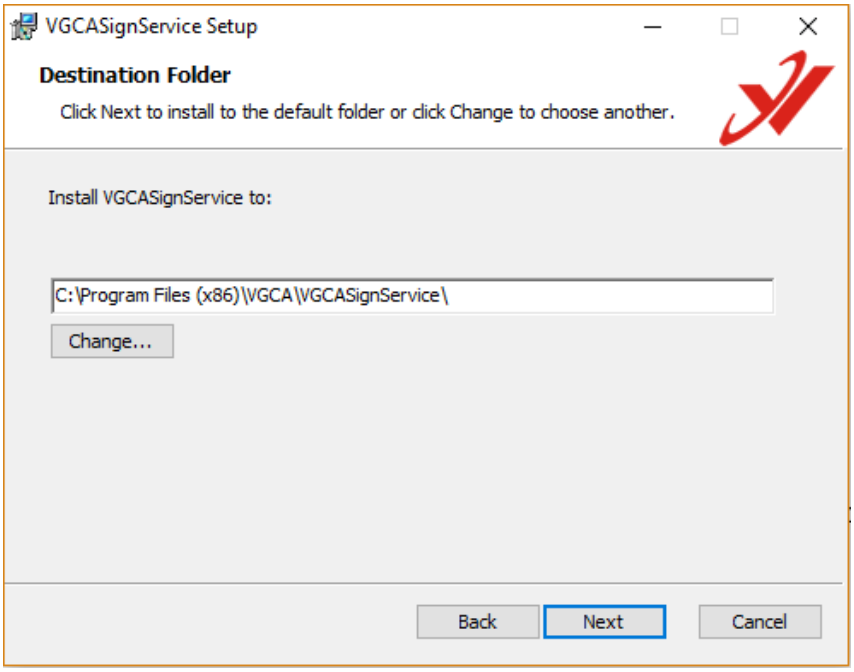


6.4 Cài đặt tool VGCASignService

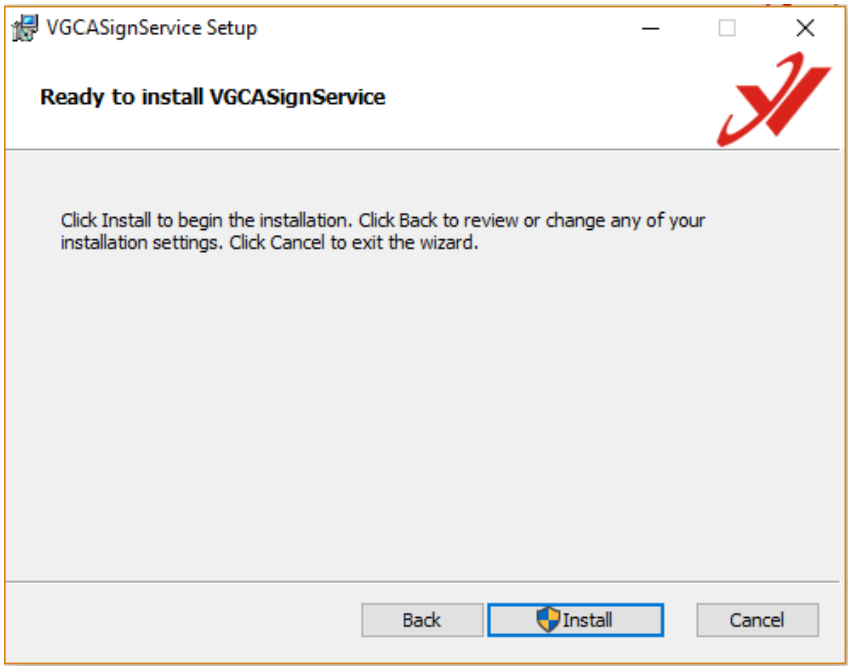
Để cài đặt tool ký số - xác thực tài liệu VGCASignService, chạy file VGCASignServiceSetup.msi



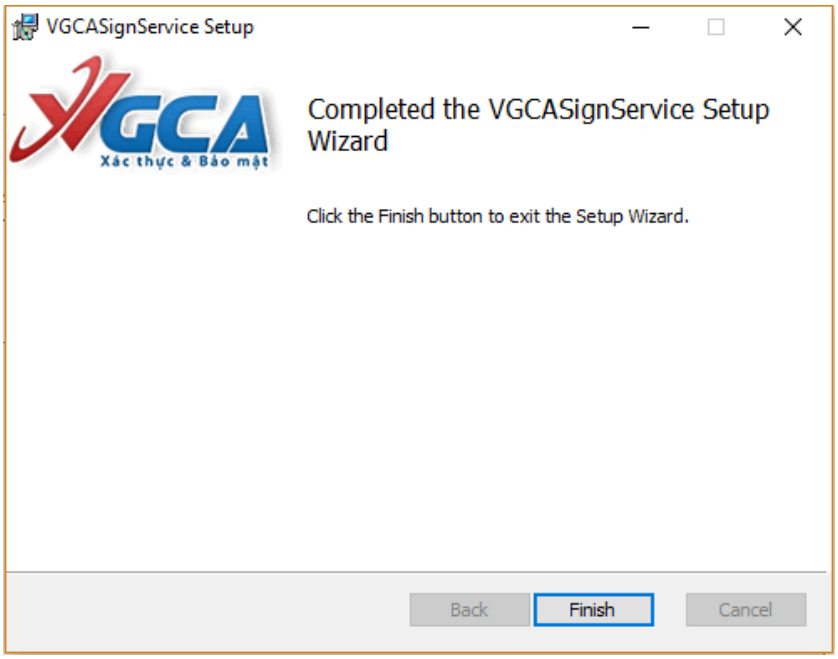
Ở cửa sổ tiếp theo cho phép bạn thay đổi đường dẫn cài đặt. Bạn chọn Next,…



Chọn Install để bắt đầu cài đặt



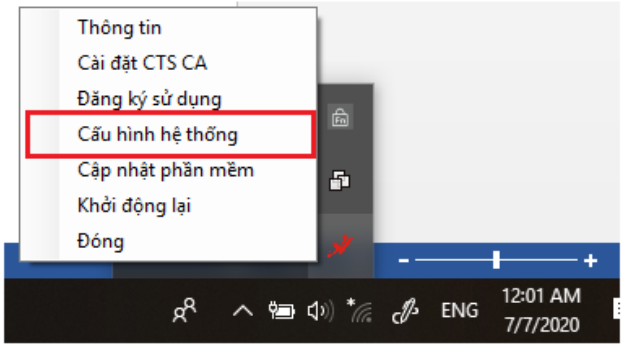
Quá trình cài đặt thành công, click chọn "Finish" để kết thúc.



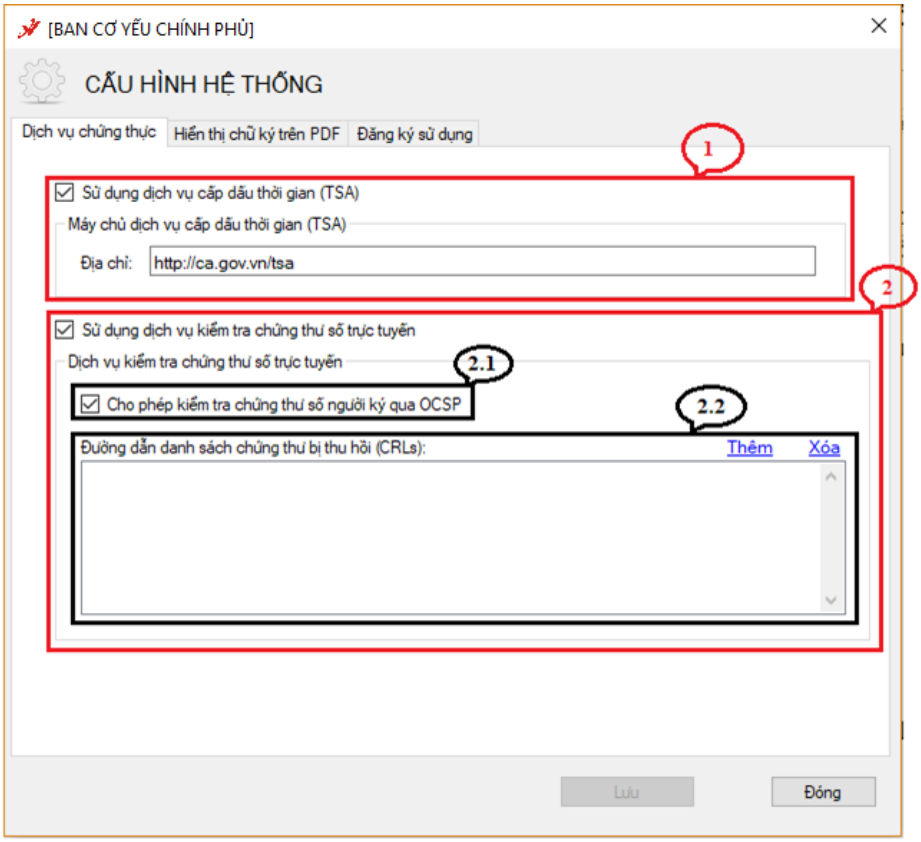
6.5 Hướng dẫn cấu hình dịch vụ chứng thực

Chú ý: Cấu hình sử dụng dịch vụ chứng thực chữ ký số của tổ chức cung cấp dịch vụ chứng thực đã được thiết lập mặc định trong phần mềm. Nếu muốn thay đổi cấu hình, người dùng thực hiện theo các bước sau:

Bước 1: Chọn chuột phải vào logo của tool VGCASignService, chọn "Cấu hình hệ thống"



Bước 2: Trên giao diện của chức năng "Cấu hình hệ thống" chọn mục "Dịch vụ chứng thực"



(1)-Cấu hình sử dụng dịch vụ cấp dấu thời gian, nhằm mục đích gắn dấu thời gian cho chữ ký. Tích chọn "Sử dụng dịch vụ cấp dấu thời gian (TSA)", nhập địa chỉ máy chủ cấp dấu thời gian vào khung Địa chỉ: http://ca.gov.vn.

(2)-Cấu hình sử dụng dịch vụ kiểm tra trạng thái thu hồi của chứng thư số. Tích chọn "Sử dụng dịch vụ kiểm tra trạng thái thu hồi của chứng thư số".

(2.1)-Tích chọn "Cho phép kiểm tra chứng thư số người ký qua OCSP" để sử dụng dịch vụ Trạng thái chứng thư trực tuyến (OCSP), mục đích là chỉ định sử dụng dịch vụ OCSP thay vì kiểm tra trong danh sách thu hồi (CRLs).

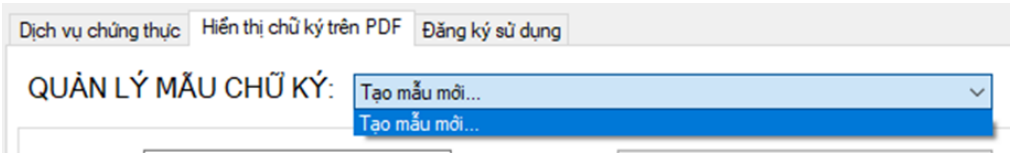
(2.2)-Thêm hoặc xóa danh sách thu hồi (CRLs)

Bước 3: Chọn “Lưu” để lưu thông tin cấu hình

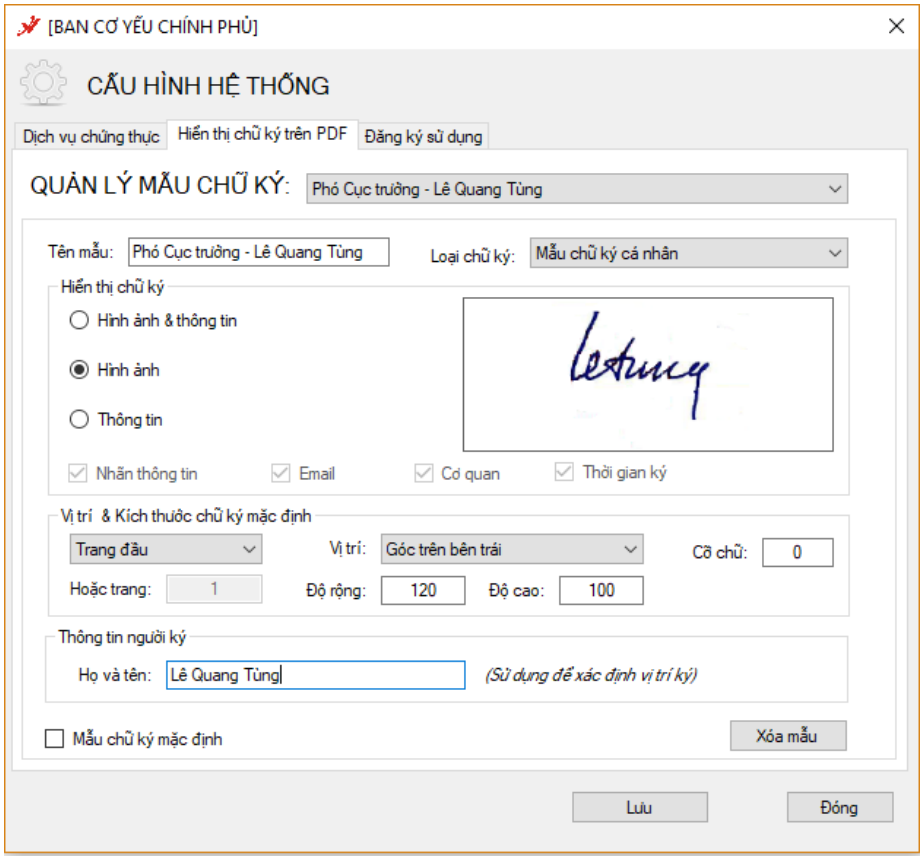
6.6 Hướng dẫn cấu hình mẫu chữ ký

a. Cấu hình form cho lãnh đạo ký số phê duyệt công văn

- Mở cấu hình Hiển thị chữ ký trên PDF
- Chọn Tạo mẫu mới…

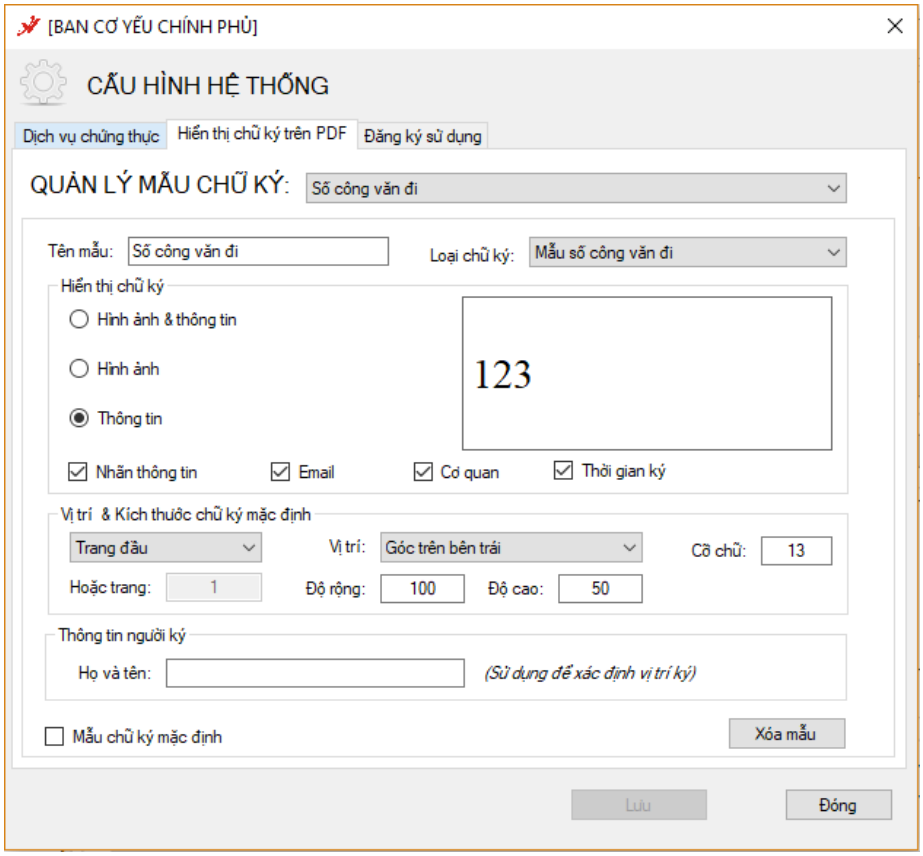


- Nhập tên mẫu lãnh đạo ký phê duyệt (Ví dụ: Phó Cục trưởng – Lê Quang Tùng)
- Chọn loại chữ ký: Mẫu chữ ký cá nhân
- Chọn hiện thị chữ ký: Hình ảnh
- Chọn ảnh chữ ký của lãnh đạo, định dạng .png bằng cách bấm chuột phải vảo hình ảnh chữ ký và chọn menu “Thay ảnh khác”
- Nhập độ rộng và độ cao của ảnh chữ ký theo đơn vị Points (1 point = 1 pixel x 96 / 72)
- Nhập tên lãnh đạo ký số công văn để tự động tìm kiếm vị trí chữ ký.
- Các thông tin khác không cần thay đổi
- Bấm Lưu để tạo và lưu mẫu.

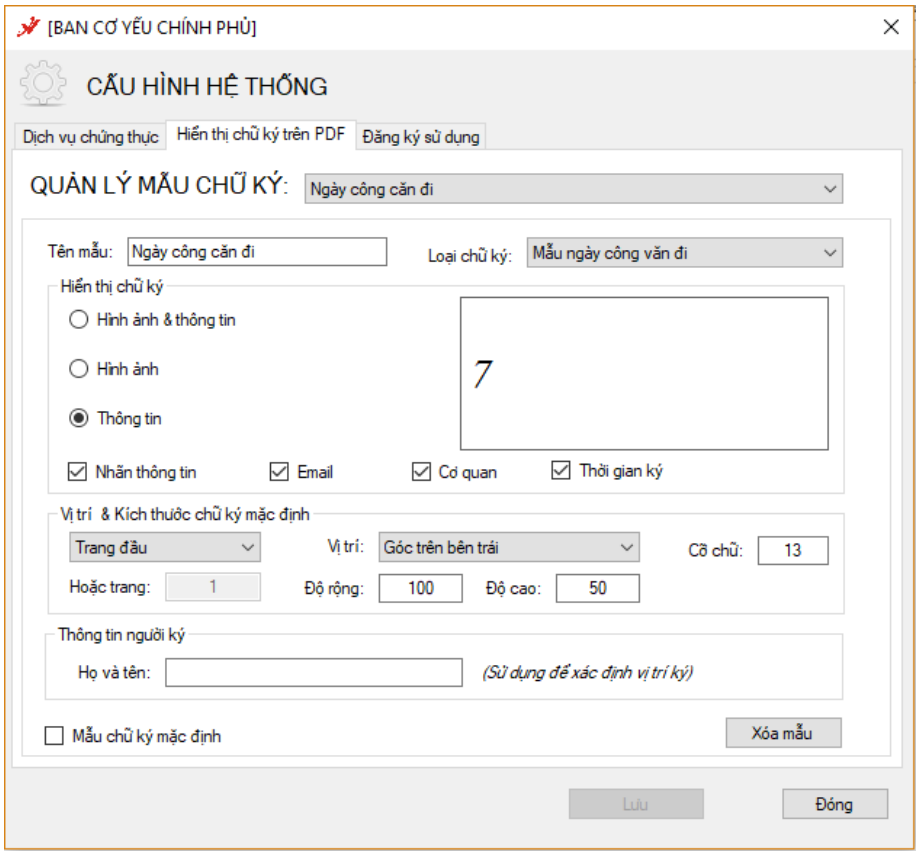


b. Cấu hình cho văn thư ký số phát hành công văn

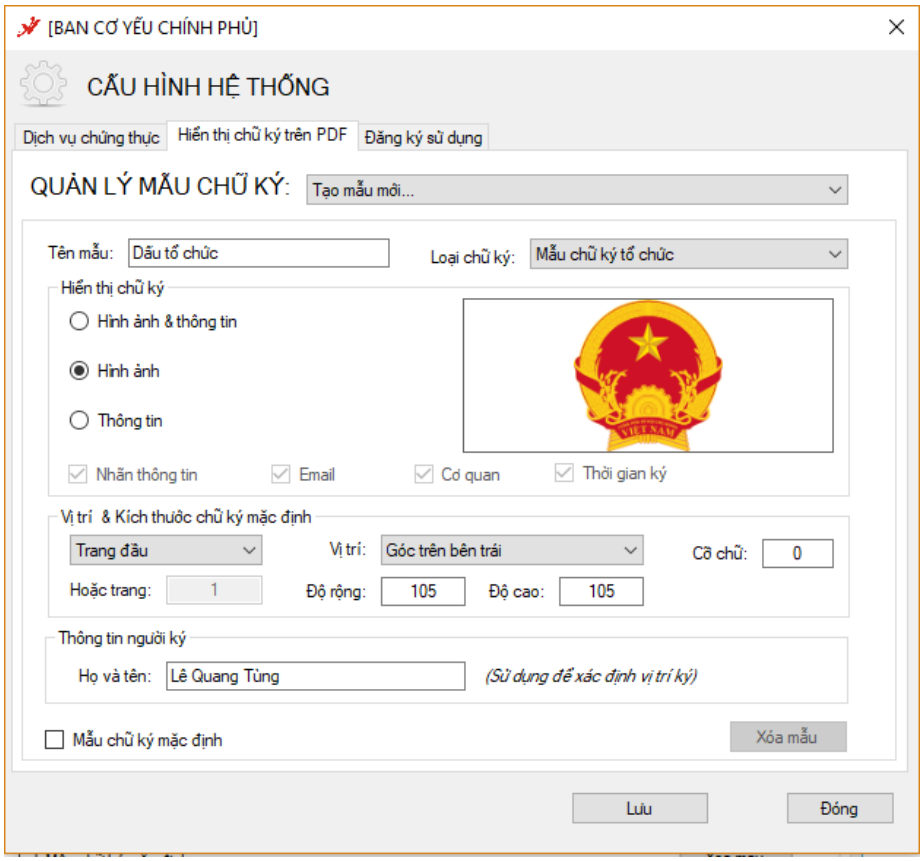
- Tạo mẫu số công văn đi:
  - Nhập tên mẫu (Ví dụ: Số công văn đi)
  - Loại chữ ký: Mẫu số công văn đi
  - Hiển thị chữ ký: Thông tin
  - Cỡ chữ: 13



- Tạo mẫu ngày công văn đi
  - Nhập tên mẫu (Ví dụ: Ngày công văn đi)
  - Loại chữ ký: Mẫu ngày công văn đi
  - Hiển thị chữ ký: Thông tin
  - Cỡ chữ : 13



- Tạo mẫu dấu tổ chức tương ứng với lãnh đạo ký số công văn
  - Tên mẫu (Ví du: Dấu tổ chức)
  - Loại chữ ký: Mẫu chữ ký tổ chức
  - Hiển thị: Hình ảnh
  - Thay ảnh dấu của đơn vị
  - Độ cao bằng độ cao của ảnh theo đơn vị Points (1px = 0.75point)
  - Độ rộng bằng độ rộng của ảnh con dấu của tổ chức theo đơn vị point
  - Nhập họ tên lãnh đạo ký số công văn.



7 Thông tin liên hệ hỗ trợ

7.1 Cục Chứng thực số và Bảo mật thông tin

Địa chỉ: Lô CN27A, Khu công nghệ cao Hòa Lạc, Hòa Lạc, Hà Nội

Điện thoai: 0243.773.8668

Email: ca@bcy.gov.vn

Website: https://ca.gov.vn