ICS 35.030
CCS L 80

GM

中华人民共和国密码行业标准

GM/T 0126—2023

HTML密码应用置标语法

HTML cryptographic application markup syntax

2023-12-04 发布　　2024-06-01 实施

国家密码管理局　　发布

目　　次

前　言

本文件按照 GB/T 1.1—2020《标准化工作导则　第 1 部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由密码行业标准化技术委员会提出并归口。

本文件起草单位：北京海泰方圆科技股份有限公司、格尔软件股份有限公司、北京小雷科技有限公司、中电科网络安全科技股份有限公司、吉大正元信息技术股份有限公司、兴唐通信科技有限公司、无锡江南信息安全工程技术中心。

本文件主要起草人：蒋红宇、柳增寿、郑强、张立廷、安晓江、王溢、罗俊、罗影、王鹏、王斌、赵丽丽、王妮娜、徐明翼。

引　言

本文件在浏览器处理的超文本置标语言中引入密码标签，通过密码标签提升网页访问的安全性。密码标签的意义在于实现网页数据交互内置的安全性。同 HTTPS 和 SSL 相比，密码标签不是工作在传输层，而是工作在应用层。

HTML 密码应用置标语法

1 范围

本文件定义了 HTML 密码标签的交互过程、密码标签格式及其解析过程和网页安全要求。

本文件适用于浏览器对网页密码标签处理。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本(包括所有的修改单)适用于本文件。

GB/T 18792—2002 信息技术 文件描述和处理语言 超文本置标语言(HTML)

3 术语和定义

下列术语和定义适用于本文件。

3.1

超文本置标语言 hyper text markup language

由 GB/T 18792—2002 所规范的一种用于描述网页的置标语言。

3.2

标签 tag

一套置标语法，用于对网页的描述。

3.3

属性 attribute

开始标签中的用等号连接的名字和值。

3.4

密码应用 cryptographic application

用于加密、解密、签名和验签密码功能的应用。

3.5

密码标签 cryptographic tag

用于完成密文、数字签名、数字证书等密码功能的网页标签。

4 缩略语

下列缩略语适用于本文件。

base64：一种 RFC4648 定义的编码(Base 64 Encoding)

ECB：电码本工作模式(Electronic Codebook Operation Mode)

HTML：超文本置标语言(Hyper Text Markup Language)

HTTP：超文本传输协议(HyperText Transfer Protocol)

ID：标识(Identity)

5 网页密码标签交互过程

5.1 交互过程

网页应用中为保证应用数据在网络传输中的安全性，应对服务器和客户端之间传递的网页内容中的特定元素进行密码保护。为此应对网页标签进行扩展，以支持加密和签名数据的传输和处理。

客户端首先应配置用户的加密/签名证书及密钥，对于网站的加密/签名证书，可事先根据域名进行配置，也可后续通过网站的密码应用网页下发。客户端应使用浏览器进行网页文件的处理。服务端应配置自身的加密/签名证书及密钥，也应预先配置用户的加密/签名证书。

支持密码标签的网页处理流程如图 1 所示。

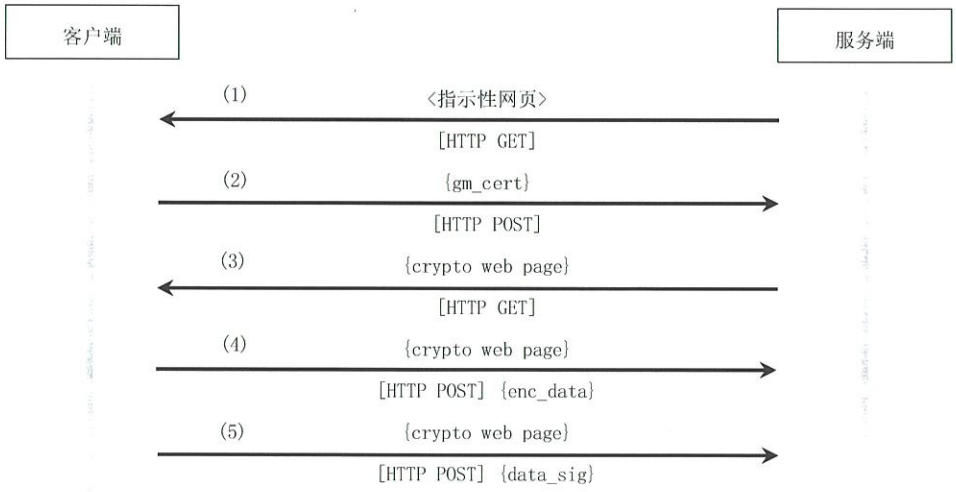


图 1 密码应用网页处理流程

在获取密码应用网页(3)之前，首先服务端根据策略判断是否向浏览器发送上传证书的指示性网页(1)，浏览器收到(1)后，会上传客户端证书(2)。在获取了密码应用网页(3)之后，如果需要上传会话密钥加密的用户数据，将执行(4)，如果需要上传用户签名的数据，将执行(5)。

网页密码标签由浏览器解析。应用签名标签时，服务端期望客户端根据标签定义进行签名操作，服务端会对签名进行验签。应用加密标签时，就是服务端期望客户端根据标签定义进行加密操作，服务端会对密文进行解密。浏览器所处理的 HTML 网页应符合 GB/T 18792—2002 的规定。

5.2 指示性网页获取

当浏览器访问密码应用网页时，服务端根据策略判断是否向浏览器发送上传证书的指示性网页(indicate web page)，指示浏览器上传客户端证书。

5.3 客户端证书上传

如果收到指示性网页，浏览器应通过 HTTP POST 方法将客户端证书(gm_cert)上传。

5.4 密码应用网页下载

浏览器应通过 HTTP GET 方法获取相应的密码应用网页(crypto web page)，网页中的内容包含

服务端证书信息、用服务端私钥签名的数据、用客户端证书公钥保护的会话密钥的数据以及用会话密钥加密的数据，浏览器应进行相应的验签和解密操作，成功则进行解析和网页显示。

5.5 加密数据上传

在密码应用网页中输入用户数据后，如果数据属于要加密的范围，浏览器应生成会话密钥并使用服务端证书公钥加密保护，然后使用会话密钥加密相应的数据，并通过 HTTP POST 方法上传加密的会话密钥和加密数据，其他部分与普通网页操作相同。

5.6 签名数据上传

在密码应用网页中输入用户数据后，如果数据属于要签名的范围，浏览器应使用客户端私钥签名相应的数据，并通过 HTTP POST 方法上传数据和签名，其他部分与普通网页操作相同。

6 密码标签格式

6.1 证书标签

开始标签：〈gm_cert name＝v1 type＝v2 value＝v3〉

结束标签：〈/gm_cert〉

在开始标签中名字为 name 的属性，v1 为其名字。

在开始标签中名字为 type 的属性，v2 为其证书类型标识串，支持的取值包括“encrypt”和“signature”。

在开始标签中名字为 value 的属性，v3 为其证书内容，证书内容使用 base64 编码。

该标签位于〈body〉标签内，供后续的加密、验签等标签使用。标签示例见附录 A。

6.2 会话密钥标签

开始标签：〈gm_sessionkey name＝v1 cert_id＝v2 value＝v3〉

结束标签：〈/gm_sessionkey〉

在开始标签中名字为 name 的属性，v1 为其名字。

在开始标签中名字为 cert_id 的属性，v2 为加密会话密钥所使用的客户端公钥证书的 ID，该 ID 为证书序列号。

在开始标签中名字为 value 的属性，v3 为其加密会话密钥内容，加密会话密钥内容使用 base64 编码。

该标签位于〈body〉标签内，其中会话密钥使用证书公钥加密，供后续的加密、解密等标签使用。

6.3 签名标签

开始标签：〈gm_sign_form〉

结束标签：〈/gm_sign_form〉

在开始标签和结束标签之间包含 input 标签用于用户的输入。

在提交事件触发后，除了提交用户 input 标签对应的 name/value 对之外，应后接数字签名项。

数字签名项的名字为 gmsign，其类型为 text，其取值字符串通过如下步骤得到。

a) 使用用户签名私钥对用户 input 标签对应的 name/value 对的字符串进行数字签名所得的签名值。

b) 对签名值进行 base64 编码的到字符串。

提交数据通过 HTTP POST 方式发送服务端。

6.4 验签标签

开始标签：〈gm_verify name=v1 cert_id=v2 alg_hash=v3 hash=v4 value=v5〉

结束标签：〈/gm_verify〉

在开始标签中名字为 name 的属性，v1 为其名字。

在开始标签中名字为 cert_id 的属性，v2 为其签名所使用的服务端公钥证书的 ID，该 ID 为证书序列号。

在开始标签中名字为 alg_hash 的属性，v3 为其杂凑算法标识串，支持的取值包括“sm3”。

在开始标签中名字为 hash 的属性，v4 为待签名的杂凑值，使用 base64 编码。

在开始标签中名字为 value 的属性，v5 为签名值，使用 base64 编码。

该标签位于〈body〉标签内，客户端对开始标签和结束标签间的网页元素的输入值使用 cert_id 对应的证书公钥进行验签操作，如果验签通过，则对网页元素进行解析和网页展示，否则应将网页元素替换成验签错误提示字符串。

6.5 图片验签标签

开始标签：〈gm_img src=v1 cert_id=v2 alg_hash=v3 value=v4 alt=v5〉

结束标签：〈/gm_img〉

在开始标签中名字为 src 的属性，v1 为其 URL，值为外部图片的地址。

在开始标签中名字为 cert_id 的属性，v2 为其签名所使用的公钥证书的 ID，该 ID 为证书序列号。

在开始标签中名字为 alg_hash 的属性，v3 为其杂凑算法标识串，支持的取值包括“sm3”。

在开始标签中名字为 value 的属性，v4 为其签名内容，签名内容使用 base64 编码。

图片验签标签是一类特殊的验签标签。该标签位于〈body〉标签内，客户端对该标签中的图片根据相应的参数进行验签操作，如果验签通过，则对图片进行解析和显示，否则应将网页元素替换成验签错误提示字符串。

6.6 加密标签

开始标签：〈gm_encrypt_form alg_encrypt=v1 mode_encrypt=v2〉

结束标签：〈/gm_encrypt_form〉

alg_encrypt 属性取值 v1 为其加密算法标识串，支持的取值包括“sm4”。mode_encrypt 的属性，v2 为其对称算法加密模式标识串，支持的取值包括“cbc”“ecb”“ofb”。

在开始标签和结束标签之间包含至少一个 input 标签用于用户的输入。

在提交事件触发后，提交用户 input 标签对应的名字/密文字符串对。

密文字符串的类型为 text，其取值字符串通过如下步骤得到。

a) 使用会话密钥对用户输入的字符串进行加密得到密文。

b) 对密文进行 base64 编码得到字符串。

提交数据通过 HTTP POST 方式发送服务端。

6.7 解密标签

开始标签：〈gm_decrypt name=v1 cert_id=v2 alg_encrypt=v3 mode_encrypt=v4 iv=v5〉

结束标签：〈/gm_decrypt〉

在开始标签中名字为 name 的属性，v1 为其名字。

在开始标签中名字为 cert_id 的属性，v2 为其加密所使用的客户端公钥证书的 ID，该 ID 为证书序列号。

在开始标签中名字为 alg_encrypt 的属性，v3 为其加密算法标识串，支持的取值包括“sm4”。

在开始标签中名字为 mode_encrypt 的属性，v4 为其对称算法加密模式标识串，支持的取值包括“cbc”“ecb”“ofb”。

在开始标签中名字为 iv 的属性，v5 为初始向量。

该标签位于〈body〉标签内，该标签将对开始标签和结束标签之间的网页元素的密文首先进行 base64 解码，然后使用会话密钥和相应的算法进行解密得到明文，并对明文进行解析和渲染。

6.8 加密签名标签

开始标签：〈gm_sigenc_form alg_encrypt＝v1 mode_encrypt＝v2〉

结束标签：〈/gm_sigenc_form〉

alg_encrypt 属性取值 v1 为其加密算法标识串，支持的取值包括“sm4”。mode_encrypt 的属性，v2 为其对称算法加密模式标识串，支持的取值包括“cbc”“ecb”“ofb”。

在开始标签和结束标签之间包含至少一个 input 标签用于用户的输入。

在提交事件触发后，提交的数据包括用户 input 标签对应的名字/密文字符串对和签名项。

密文字符串的类型为 text，其取值字符串通过如下步骤得到。

a) 使用会话密钥对用户输入的字符串进行加密得到密文。

b) 对密文进行 base64 编码得到字符串。

签名项的名字为 gmsign，类型为 text，其取值字符串通过如下步骤得到。

a) 使用用户签名私钥对用户 input 标签对应的 name/value 对的字符串散列值进行数字签名所得的签名值。

b) 对签名值进行 base64 编码的到字符串。

提交数据通过 HTTP POST 方式发送服务端。

待加密数据格式为 4 字节长度连接用户输入，若不为分组长度的整数倍，则补 0 至分组整数倍。

6.9 验签解密标签

开始标签：〈gm_decrypt_verify name＝v1 cert_enc_id＝v2 cert_sig_id＝v3 alg_hash＝v4 alg_encrypt＝v5 mode_encrypt＝v6 sign_value＝v7〉

结束标签：〈/gm_decrypt_verify〉

在开始标签中名字为 name 的属性，v1 为其名字。

在开始标签中名字为 cert_enc_id 的属性，v2 为其加密所使用的客户端公钥证书的 ID，该 ID 为证书序列号。

在开始标签中名字为 cert_sig_id 的属性，v3 为签名所使用的服务端公钥证书的 ID，该 ID 为证书序列号。

在开始标签中名字为 alg_hash 的属性，v4 为其杂凑算法标识串，支持的取值包括“sm3”。

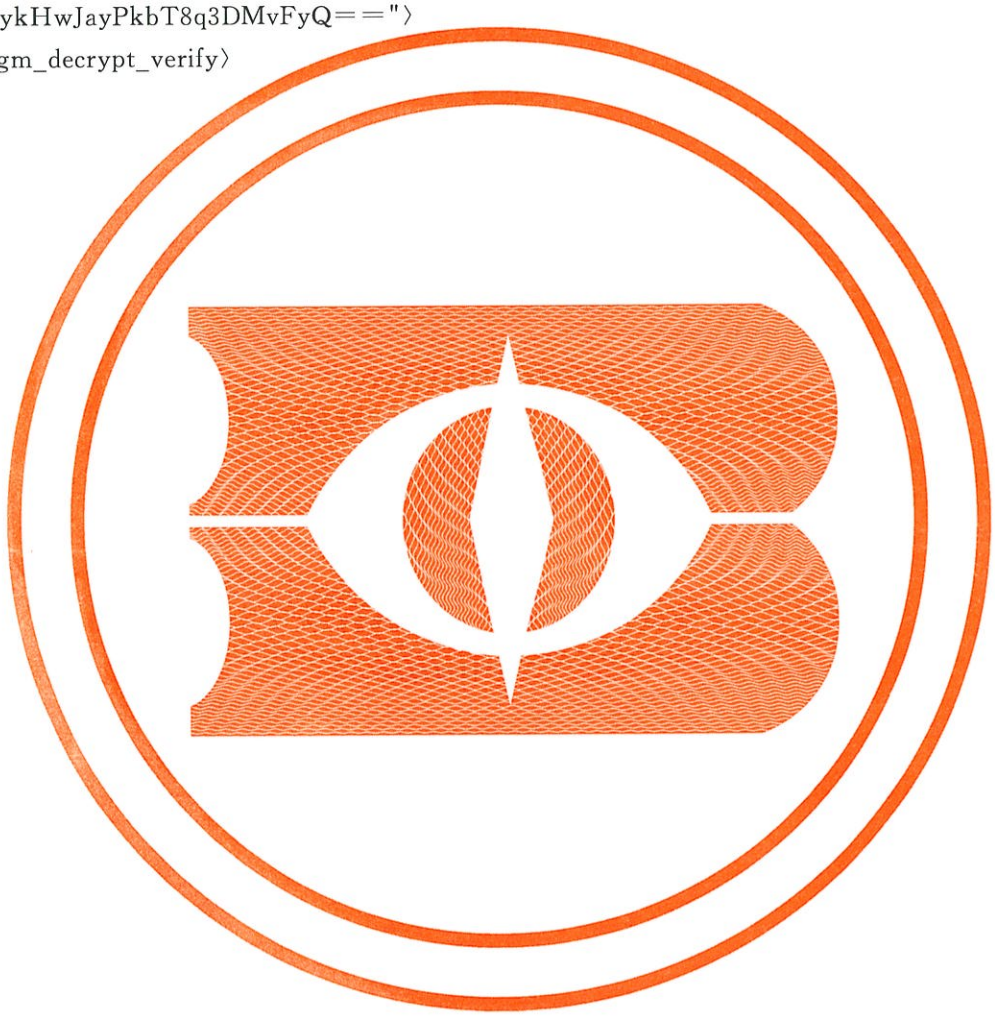
在开始标签中名字为 alg_encrypt 的属性，v5 为其加密算法标识串，支持的取值包括“sm4”。

在开始标签中名字为 mode_encrypt 的属性，v6 为其对称算法加密模式标识串，支持的取值包括“cbc”“ecb”“ofb”。

在开始标签中名字为 sign_value 的属性，v7 为其对密文的签名内容，签名内容使用 base64 编码。

首先通过对开始标签和结束标签之间的密文进行验签操作，如果验签不通过，将网页元素替换成验签错误提示字符串，通过则使用解密算法对密文进行解密得到明文，然后对明文进行解析和网页显示。

〈gm_decrypt_verify
name="gm_decrypt_verify"
cert_enc_id="0095146031"
cert_sig_id="0095146031"
alg_hash="sm3"
alg_encrypt="sm4"
mode_encrypt="ecb"
sign_value="/fCG/XkQ1PRyUdg2nRwYZ53JPJwLDe75i9UKQsokg0x618zcLWPKFMkotyEY
PIyrGpykHwJayPkbT8q3DMvFyQ==">
〈/gm_decrypt_verify〉



参 考 文 献

[1] RFC 4648 The Base16, Base32 and Base64 Data Encodings