云核客户端安全输入系统
（鸿蒙版）
用户手册

北京云核网络技术有限公司

2025 年 1 月 10 日

文档修改记录

版本	内容	编写人	编写日期	审核人	审核日期
0.1	初版	changbf	2024-04-10	luoyong	
1.0.1	添加接口	changbf	2024-06-30		
1.0.2	添加定制化接口	changbf	2024-10-17		
1.0.3	添加国际、国密算法	changbf	2024-10-20		
1.0.4	添加防重放，可通过属性设置	changbf	2024-10-21		
1.0.5	添加输入框属性	changbf	2024-11-14		
1.0.6	添加安全键盘属性	changbf	2024-11-20		
1.1.0	正式发布文档	changbf	2024-12-04		
1.1.1	增加 controller，提供 webview 使用	changbf	2024-12-06-		
1.1.2	增加-fstack-protector 选项	changbf	2025-01-04-		
1.1.3	修复全键盘点击完成无法关闭	changbf	2025-01-10		

目　录

1.引言

1.1 编写目的

“云核客户端安全输入系统”是保护用户敏感信息输入的重要手段，能对客户端操作系统进行安全增强，可大幅提升“木马”程序非法获取用户敏感输入信息“成本”的软件集合。

“云核客户端安全输入系统” 的鸿蒙版由安全输入框模块和安全键盘核心实现模块两部分组成。开发人员通常不直接调用安全键盘核心实现模块，而是主要通过使用鸿蒙 SDK 调用安全输入框模块来实现与自身业务系统的整合。本《用户手册》中将以一个典型的开发部署过程为例，为开发人员展示安全输入框的调用方法及注意事项，使开发人员能够独立应用“云核客户端安全输入系统”。

1.2 背景知识及参考资料

假定开发人员对下列技术有一定的理解：

技术	有关内容
鸿蒙 SDK	鸿蒙 ETS 的基本知识
NAPI-node	实现 ArkTS/TS/JS 和 C/C++之间的交互

1.3 使用环境

鸿蒙星河版。

2.安全输入系统概述

2.1 系统构成

云核客户端安全输入系统鸿蒙版本以 HAR 格式提供，由两部分构成：

安全键盘核心实现模块：由 C++实现的 SO 动态库。

安全输入框模块：由 ETS 实现。

2.2 功能特点

- 防止 Hook 类木马程序攻击。
- 防止网络传输泄密。

- 支持 1024、2048 位 RSA 密钥
- 支持 PKCS1 及加密机自定义填充
- 支持时间戳，防止重放攻击

- 程序防破解保护

2.3 技术特点

- 用户输入的敏感信息不在界面框中保存，而是通过加密后存放在安全键盘输入核心动态库中，有效防止了敏感信息泄漏。
- 安全键盘样式 TextInput 开发，核心均由安全键盘输入核心动态库实现。
- 支持自定义的公钥形式的加密密钥，密文已转换为 big-endian。
- 支持自定义属性，按键无序、最大/小长度等。

2.4 接口描述

2.4.1 引用

对象	说明
CTextInputProperty	安全输入框属性
CKeyboardAttribute	安全输入键盘属性
EnumKeyboardType	键盘类型 0 QWERTY 全键盘 1 NUMBER 数字键盘 2 MONEY 含点的数字键盘 3 IDCARD 含 X 的数字键盘 MONEY、IDCARD 需定制化开发
EnumKeyboardStyle	键盘式样，在全键盘时生效，默认为小写字母键盘。 QWERTY_ALPHA_LOWER 小写字母键盘 QWERTY_ALPHA_UPPER 大写字母键盘 QWERTY_SYMBOL_1 字符键盘 QWERTY_SYMBOL_2 字符键盘 NUMBER 数字键盘 可通过键盘中的按键进行切换，字符键盘 1/2 为 UI 定制化时预留,通用版不区分
EnumKeyClickMode	按键点击时的效果，默认点击回显 CHANGE 点击时回显 NOCHANGE 点击时无状态，更安全
EnumContentType	输入内容类型

	ANY 输入内容不要求 NUM 必须含有 数字 LETTER 必须含有 字母 NUM_AND_LETTER 必须含有 数字和字母 PUNCT 必须含有 符号 NUM_AND_PUNCT 必须含数 字、符号 LETTER_AND_PUNCT 必须含有 字母、符号 NUM_LETTER_PUNCT 必须含有 数字、字符、符号
EnumDictMatchType	设置字典时，匹配方式 FULL_MATCH 全文匹配 DICT_INCLUDE_PASS_MATCH 字典包含密码匹配 PASS_INCLUDE_DICT_MATCH 密码包含字段匹配
EnumFinishMode	点击完成时，是否关闭键盘 CLOSE 自动关闭 NOT_CLOSE 不关闭，业务逻辑可通过 provider 中的 controller 关闭
EnumPollState	监听键盘状态 KEYBOARD_CREATE 键盘显示 KEYBOARD_OPEN 键盘显示 KEYBOARD_CLOSE 键盘关闭 KEYBOARD_FINISH 键盘完成 点击完成按钮事件 CONTENT_UPDATE 输入内容变化
EnumRandomType	键盘乱序方式 NULL 不乱序 ALL，乱序所有内容，字母按所在行乱序 NUMBER 仅乱序数字键盘
EnumSwitchMode	全键盘切换模式，定制键盘预留 BETWEEN_ALPHA_SYMBOL 字母符号切换 BETWEEN_ALPHA_NUMBER 字母数字切换 SWITCH_ALPHA_SYMBOL_NUMBER 字母数字符号切换，通用版本仅此模式
EnumEnvironment	多语言版本，默认简体中文，其他需定制化开发
CiProtectProvider	安全输入 Provider，提供键盘对象方法
CiProtectController	密码控件 Controller，用于 WEB 等环境使

	用
CTextInput	安全输入框
CWebKeyboard	安全输入键盘，可在应用中单独使用
CAloneKeyboard	安全输入键盘，可在应用中单独使用

2.4.2 安全输入框属性

```
export class CTextInputProperty {
  placeholderColor(value: ResourceColor): CTextInputProperty
  placeholderFont(value: Font): CTextInputProperty
  height(value: Length):  CTextInputProperty
  fontSize(value: Length): CTextInputProperty
  backgroundColor(value: ResourceColor): CTextInputProperty
  margin(value: Margin | Length): CTextInputProperty
  padding(value: Padding | Length): CTextInputProperty
  maskCode(value: string): CTextInputProperty
  caretColor(value: string): CTextInputProperty
  defaultFocus(value: boolean): CTextInputProperty
  focusable(value: boolean): CTextInputProperty
  key(value: string): CTextInputProperty
  id(value: string): CTextInputProperty
  bolder(value: BorderOptions): CTextInputProperty
}
```

方法说明：

方法	说明
placeholderColor(value: ResourceColor)	设置 placeholder 颜色，可以链式设置
placeholderFont(value: Font)	设置 placeholder 字体大小，可以链式设置。默认值为{size: 16}
fontSize(value: Length)	设置输入框输入内容字体大小，可以链式设置。默认为 16fp
height(value: Length)	设置输入框高度，可以链式设置．默认为 48vp
backgroundColor(value: ResourceColor)	设置输入框背景颜色，可以链式设置。默认白色
margin(value: Margin | Length)	设置 margin，可以链式设置
padding(value: Padding | Length)	设置 padding，可以链式设置
maskCode(value: string)	设置掩码值，可以链式设置。默认'*'
caretColor(value: ResourceColor)	设置光标颜色
defaultFocus(value: boolean)	设置当前组件是否为当前页面上的默认焦点

focusable(value: boolean)	设置当前组件是否可以获焦
key(value: string)	组件的唯一标识，唯一性由使用者保证
id(value: string)	组件的唯一标识，唯一性由使用者保证
bolder(value: BorderOptions)	边框属性

2.4.3 安全输入键盘属性

```
export class CKeyboardAttribute {
  type(type: EnumKeyboardType): CKeyboardAttribute
  style(style: EnumKeyboardStyle): CKeyboardAttribute
  keyClick(state: EnumKeyClickMode): CKeyboardAttribute
  secure(secure: boolean): CKeyboardAttribute
  acceptRegExp(value: string): CKeyboardAttribute
  maxLength(length: number): CKeyboardAttribute
  minLength(length: number): CKeyboardAttribute
  keyRandomType(randomType: EnumRandomType): CKeyboardAttribute
  vibrator(vibrator: boolean): CKeyboardAttribute
  contentType(contentType: EnumContentType): CKeyboardAttribute
  switchMode(switchMode: EnumSwitchMode): CKeyboardAttribute
  publicKey(appPubKey: string, encryptPubKey?: string): CKeyboardAttribute
  algorithmCode(code: string): CKeyboardAttribute
  calcFactor(factor: string): CKeyboardAttribute
  hashRandom(random: string): CKeyboardAttribute
  challengeCode(code: string): CKeyboardAttribute
  dictFilter(dict: string, matchType: EnumDictMatchType): CKeyboardAttribute
}
```

参数说明：

属性	说明
type(type: EnumKeyboardType)	安全键盘类型， 初始化时指定。 默认 0：全键盘
style(style: EnumKeyboardStyle)	安全键盘键盘风格，初始化时指定。当键盘类型为全键盘时生效。默认 QWERTY_ALPHA_LOWER
keyClick(state: EnumKeyClickMode)	设置按键点击状态，默认点击反显

secure(secure: boolean)	是否防截屏，默认为 true，需要 ohos.permission.PRIVACY_WINDOW 权限
finishClick(value: EnumFinishMode)	设置点击完成按键是否关闭。默认为关闭。如果设置不关闭，可在 PollCallback 回调中处理关闭事件
acceptRegExp(accepts: string)	设置可以接受的字符。初始化时指定。accepts 为 deelx 正则表达式。默认“[:print:]+”，见 2.4.6 章节内容介绍
maxLength(length: number)	设置最大长度，默认 12。当输入内容达到最大长度内容时，将不能再输入
minLength(length: number)	设置最小长度，默认 6。verify()方法会最小值判断输入内容是否满足要求，见 verify 返回值。
keyRandomType(randomType: EnumRandomType)	设置键盘乱序方式，默认为 ALL
vibrator(vibrator: boolean)	设置点击按键震动，需要 ohos.permission.VIBRATE 权限
contentType(contentType: EnumContentType)	设置输入内容要求，默认为 ANY
switchMode(switchMode: EnumSwitchMode)	设置切换模式，暂无用，定制化 UI 可使用
publicKey(appPubKey: string, encryptPubKey?: string)	设置应用平台公钥和加密平台公钥，不对接加密机只需要设置应用平台公钥，应用平台公钥由业务提供，使用 ccp-security.jar 解密，加密平台公钥由加密机提供，使用时请参考示例
publicKeyAppModulus(pubKey: string)	单独设置应用平台公钥，使用 ccp-security.jar 解密（单层算法定制自行解密），RSA 模数或者 SM2 公钥，16 进制编码格式
publicKeyModulus(pubKey: string)	单独设置加密平台公钥， RSA 模数 16 进制编码格式，对接加密机，由加密机提供公钥
publicKeyECC(eccX: string, eccY: string)	单独设置加密平台公钥，仅支持国密，分别为 SM2 公钥的 x 和 y 坐标，与 publicKeyModulus 相对应
algorithmCode(code: string)	设置算法 code，多种算法进行切换，仅多种算法生效，参考额外提供的算法文档，只有一种算法时不需要设置
calcFactor(factor: string)	定制特殊算法使用，参考额外的算法文档 设置因子，X9.8 使用。factor 格式： 0：不变换格式 1：不带主账号的 X9.8 格式 2+主账号：带主账号的 X9.8 格式
hashRandom(random: string)	设置 hashRandom，定制化特殊算法使用 "1:"+salt; //16 位 MD5 结果 "2:"+salt //32 位 MD5 结果
challengeCode(code: string)	设置挑战码，定制化特殊算法使用
dictFilter(dict: string, matchType:	设置密码字典和匹配类型

EnumDictMatchType)	
switchMode(switchMode: EnumSwitchMode)	暂无用，保留接口
switchEnviron(environ: EnumEnvironment)	多语言设置，定制化 UI 的版本使用
switchLogo(logoIndex: number)	切换内置多种 logo，定制化 UI 的版本使用，最多支持 3 个
darkMode(mode: boolean)	暗黑模式，定制化 UI 的版本使用

2.4.4 CiProtectProvider

方法	说明

constructor(callback?: PollCallback)	初始化时回调，接收键盘状态
	export interface PollCallback {
	(state: EnumPollState, length: number, degree: string): void
	}

getValue(timeStamp:string):string | number	获得输入加密后的输入内容或错误码。timestamp:时间戳，单位：毫秒/秒，一般使用毫秒。 当区分一次加密和二次加密，getValue()为一次加密，getPinValue()为二次加密。默认为云核防重放密文格式，使用 ccp-security.jar 进行解密
getPinValue(timestamp:string):string | number	获得输入加密后的输入内容 或 错误码。timestamp:时间戳，单位：毫秒/秒，一般使用毫秒。 当区分一次加密和二次加密，getValue()为一次加密，getPinValue()为二次加密。当使用对接加密机时，使用二次加密，最外层密文有 ccp-security 解密，解密出来的密文由加密机转加密
setPublicKey(appPublicKey:string, encryptPublicKey: string): number	和 attribute 中 publicKey 接口相同
verify(): number	验证输入内容，返回值： 0：合法 -1：内容为空 -2：输入小于最小长度 -3：输入字符不可接受 输入内容不符合 accept -4：简单密码 输入的密码为简单密码，需定制 -5：输入的密码是字典密码，匹配上设置的字典 -6：错误的校验类型，不符合 contentType
clear(): void	清除输入内容

getLength(): number	获得输入内容长度
switchEnviron(environ: EnumEnvironment) : void	多语言设置，定制化 UI 的版本使用
switchLogo(logoIndex: number): void	切换内置多种 logo，定制化 UI 的版本使用，最多支持 3 个
darkMode(mode: boolean): void	暗黑模式，定制化 UI 的版本使用
getVersion(): string	获得版本，保留接口，待实现
getDegree(): string	获得输入内容强度，通常为：S/M/E 返回值： S：强 H：高 M：中 W：弱 F：禁止使用 E：内容为空
getLastError(): string	获得最后一次的错误信息，当接口异常时，通过此接口获取详细的错误信息
setCalcFactor(String factor): void	设置因子，X9.8 算法使用。 factor 格式： 0：不变换格式 1：不带主账号的 X9.8 格式 2+主账号：带主账号的 X9.8 格式
getMeasureValue(): string	获取密码控件输入内容的测量值，如果两个控件的测量值一样，说明密码是相等的

2.4.5 CiProtectController

方法	说明

constructor(callback?: PollCallback)	初始化时回调，接收键盘状态
	export interface PollCallback {
	(state: EnumPollState, length: number, degree: string): void
	}

getInstance()	获取 Controller 实例

create(windowStage: window.WindowStage | null, kbData: ControllerKeyboardData): string	创建键盘
	export interface ControllerKeyboardData {
	keyboardId: string
	attribute: CKeyboardAttribute
	}

show(keyboardId: string, callback?: ControllerCallback<ControllerChangeData>)	根据 create 的键盘 ID 显示键盘
	
	export interface ControllerCallback<T> {
	(data?: T): void
	}
	
	export interface ControllerChangeData {
	id: string,
	state: number,
	length: number,
	degree: string,
	height: number,
	mask: string
	}

dismiss(keyboardId: string)	根据 ID 关闭键盘
destroy(keyboardId: string)	根据 ID 摧毁资源
getKeyboardHeight(keyboardId: string):number	获取键盘高度
getMeasureValue(keyboardId: string):string	获取测量值，通过测量值可比较两个键盘输入内容是否相同
getPinValue(keyboardId: string , timestamp: string):string	同 provider 中的方法
getValue(keyboardId: string, timestamp: string):string	同 provider 中的方法
verify(keyboardId: string):number	同 provider 中的方法
clear(keyboardId: string)	同 provider 中的方法
publicKeyModulus(keyboardId: string, publicKey: string)	设置加密平台公钥，和 CKeyboardAttribute 中的接口相同
publicAppModulus(keyboardId: string, publicKey: string)	设置应用平台公钥，和 CKeyboardAttribute 中的接口相同
publicKeyECC(keyboardId: string, eccX: string, eccY)	设置加密平台公钥，和 CKeyboardAttribute 中的接口相同
lastError(keyboardId: string):string	获取错误信息，当调用 getPInValue 和 getValue 没有密文时，调用此接口查看错误原因
getProvider(keyboardId: string):CiProtectProvider	获得 provider，可通过 provider 调用键盘的各种剪口，见 CiProtectProvider 中的说明

参考示例中的 SecurityInput，完成 WEB 上的键盘引用。

2.4.6 正则表达式

“云核客户端安全输入系统”中使用了 DEELX 正则表达式引擎。DEELX 是一个在 C++环境下的与 Perl 兼容的正则表达式引擎，是 RegExLab 开展的一个研究开发项目。

- **简单的转义字符**

一些不便书写的字符，比如换行符，制表符等，使用\n，\t 来表示。另外有一些标点符号在正则表达式中，被定义了特殊的意义，因此需要在前面加 "\" 进行转义后，匹配该字符本身。

DEELX 中的转义字符：

转义符	说明
\a	响铃符 =\x07
\f	换页符 =\x0C
\n	换行符 =\x0A
\r	回车符 =\x0D
\t	制表符 =\x09
\v	垂直制表符 =\x0B
\e	ESC 符 =\x1B
\x20	使用两位十六进制表示形式，可与该编号的字符匹配
\u002B	使用四位十六进制表示形式，可与该编号的字符匹配
\x{20A060}	使用任意位十六进制表示形式，可与该编号的字符匹配

图表 1DEELX 中的转义字符

在 DEELX 中被定义了特殊的意义，因而需要在前面添加 ″\″ 来匹配该字符本身的标点符号：

字符	说明
^	匹配输入字符串的开始位置。要匹配 "^" 字符本身，请使用 "\^"
$	匹配输入字符串的结尾位置。要匹配 "$" 字符本身，请使用 "\$"
( )	标记一个子表达式的开始和结束位置。要匹配小括号，请使用 "\(" 和 "\)"
[ ]	用来自定义能够匹配 '多种字符' 的表达式。要匹配中括号，请使用 "\[" 和 "\]"
{ }	修饰匹配次数的符号。要匹配大括号，请使用 "\{" 和 "\}"
.	匹配除了换行符（\n）以外的任意一个字符。要匹配小数点本身，请使用 "\."

?	修饰匹配次数为 0 次或 1 次。要匹配 "?" 字符本身，请使用 "\?"
+	修饰匹配次数为至少 1 次。要匹配 "+" 字符本身，请使用 "\+"
*	修饰匹配次数为 0 次或任意次。要匹配 "*" 字符本身，请使用 "*"
|	左右两边表达式之间 "或" 关系。匹配 "|" 本身，请使用 "|"

图表 2DEELX 特殊意义字符

- **字符集合**

可以匹配 ″多个字符″ 其中任意一个字符的正则表达式。虽然是 ″多个字符″，但每次只能匹配其中一个。

字符集合	说明
.	小数点可以匹配除了换行符（\n）以外的任意一个字符
\w	可以匹配任何一个字母或者数字或者下划线
\W	W 大写，可以匹配任何一个字母或者数字或者下划线以外的字符
\s	可以匹配空格、制表符、换页符等空白字符的其中任意一个
\S	S 大写，可以匹配任何一个空白字符以外的字符
\d	可以匹配任何一个 0~9 数字字符
\D	D 大写，可以匹配任何一个非数字字符
[:alpha:]	POSIX 格式，可以匹配任何一个字母
[:^alpha:]	POSIX 否定格式，可以匹配任何一个字母以外的字符

图表 3DEELX 字符集合

DEELX 支持的 POSIX 字符集合定义：

POSIX 字符集合	说明
[:alnum:]	任何一个字母或数字（A-Z，a-z，0-9）
[:alpha:]	任何一个字母（A-Z，a-z）
[:ascii:]	任何一个 ASCII 范围内字符（\x00–\x7F）
[:cntrl:]	任何一个控制字符（\x00–\x1F，\x7F）
[:digit:]	任何一个数字（0–9）
[:print:]	任何一个可显示的 ASCII 字符（\x20–\x7E）
[:space:]	任何一个空白字符（\x09–\x0D，\x20）
[:graph:]	任何一个可显示的 ASCII 字符，不包含空格（\x21–\x7E）
[:lower:]	任何一个小写字母（a–z）
[:punct:]	可显示字符 [:print:] 中除去字母数字 [:alnum:]
[:upper:]	任何一个大写字母（A–Z）
[:xdigit:]	任何一个十六进制数字（0-9，A-F，a-f）

[:blank:]	空格或者制表符（\x20，\x09）

图表 4DEELX 支持的 POSIX 字符集合

所有的 POSIX 字符集合，与[:^alpha:]类似 ，当[:之后为^时，表示相应字符集合之外的字符。

更详细的 DEELX 正则表达式语法可以在互联网上找到。

- **匹配方式**

“云核客户端网银安全服务”从 3.72.0.0 版本（含）开始通过在正则表达式前面添加前缀方式，支持两种匹配方式：

1、 精准匹配

所谓精准匹配，就是要求正则表达式从文本开始位置刚好匹配到文本结束位置，表示输入内容是可以被安全服务所接受。精准匹配用 Exact:前缀表示。

比如：

"accept-regexp":"Exact:[:alnum:]"

代表采用精准匹配方式，输入内容必须由字母或数字（A - Z, a - z, 0 - 9）构成。

精准匹配方式是默认方式，所以 Exact:前缀可以省略，即：

"accept-regexp":"[:alnum:]"

2、 查找匹配

所谓查找匹配，就是从 0 开始，从左向右查找匹配符合表达式的内容，如果找到说明匹配成功，表示输入内容是可以被安全服务所接受。查找匹配用 Inexact:前缀表示。

比如：

"accept-regexp":"Inexact:(\\D)(\\d{0,})(\\D)"

代表采用查找匹配方式，输入内容必须含有两个单独或连续的字母或符号。

注意：(\D)(\d{0,})(\D) 由于 \ 在 Json 中为转义字符，需要写成 ″(\\D)(\\d{0,})(\\D)″。

3.开发流程概述

3.1 设置属性

有两种属性，一个是输入框属性，一个是键盘属性：

1. 安全输入框属性：

```
property: CTextInputProperty = new CTextInputProperty()
  .placeholderColor('#99182431')
  .height('48vp')
  .fontSize('16fp)
  .backgroundColor(Color.White)
  .margin({ top: '10vp'})
  .padding({ left: 12})
  .maskCode('#')
```

2. 安全键盘属性

```
attribute: CKeyboardAttribute = new CKeyboardAttribute()
  .type(EnumKeyboardType.QWERTY)
  .style(EnumKeyboardStyle.NUMBER)
  .acceptRegExp('[:print:]+')
  .contentType(EnumContentType.LETTER)
  .publicKey(this.publicKey)
  .algorithmCode('003')
```

3. provider

```
provider: CTextInputProvider = new CTextInputProvider(this.pollCallback)
```

3.2 引入输入框

```
CTextInput({
  placeholder: $r('app.string.password'),
  property: this.property,
  attribute: this.attribute,
  provider: this.provider
  textAlign: TextAlign.Start,
  focusableProp:true,
  defaultFocusProp: false
})
```

4. 服务端 API 简介

客户端与服务端采用密文进行通讯，可以实现用户敏感信息（如：PIN）从浏览器到后台主机的全程保护。服务端程序主要负责解密及时间戳验证（防止重放攻击）。

JAVA 解密包为：ccp-security.jar

com.cloudcore.ccp.enter.CloudCorePinConvertor 接口说明：

函数	说明	备注
void setTimeout(long timeout)	设置超时时间	单位：分钟
String convert(String ciphertext)	rsa-rc4 数字信封解密函数，私钥在 jar 的 isecurity.properties 文件中，m 表示私钥的模数，ve 表示私钥的指数。	ciphertext：base64 编码的密文 返回值：明文 注意： 防重放因子类型须为: timestamp； 待加密内容变换类型须为: "anti-replay-colon"或"anti-replay-degree-colon"； 其他情况，服务端需要自行实现。
String convert(String ciphertext, String modulus, String exponent)	RSA 密文的解密函数，需要传入私钥。	ciphertext：base64 编码的密文 modulus：16 进制编码的私钥模数 exponent：16 进制编码的私钥指数 返回值：明文 注意： 防重放因子类型须为: timestamp； 待加密内容变换类型须为: "anti-replay-colon"或"anti-replay-degree-colon"；

		其他情况，服务端需要自行实现。
String convertSM2(String ciphertext, String privKey)	SM2 密文的解密函数，需要传入私钥。	ciphertext：base64 编码的密文 privKey：16 进制编码的私钥 返回值：明文 注意： 防重放因子类型须为:timestamp； 待加密内容变换类型须为:"anti-replay-colon"或"anti-replay-degree-colon"； 其他情况，服务端需要自行实现。

解密包函数说明

com.cloudcore.ccp.enter.SecurityHandler 接口说明：

函数	说明	备注
Map<String, String> generatorKeyPair(int keyLength, String seed)	生成 RSA 密钥对	keyLength：秘钥长度，可传入 1024、2048。 seed：随机数种子。 返回值：Map 的 KEY 字段有： PublicExponent :公钥指数 PublicModules :公钥模数 PrivateExponent:私钥指数 PrivateModulus :私钥模数
Map<String, String> generatorSM2KeyPair(String seed)	生成 SM2 密钥对	seed：随机数种子。 返回值：Map 的 KEY 字段有： PublicX：公钥 X PublicY：公钥 Y PrivateKey：私钥

密钥对生成函数说明