Cómo Firmar un TED y no morir en el intento

Capacitación Conceptual

V 9.9.3

Agenda

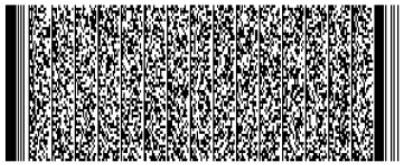
- Contexto
- Antecedentes
- ¿Cómo se aplica?
- ¿Dónde están las llaves?
- Proceso Final

CONTEXTO

¿Qué es el TED?

<DTE>

<TED>



TED - Timbre Electrónico del DTE

NODO - TED

RE - RUT EMISOR

TD - TIPO DE DOCUMENTO

F - FOLIO

FE - FECHA EMISION

RR – RUT RECEPTOR

RSR – RAZON SOCIAL RECEPTOR

MNT – MONTO NETO

IT1 – GLOSA DETALLE 1

CAF

TSTED – TimeStamp TED

FRMT – Firma TED

VOLVER

CAF – Código de Autorización de Folios

RUT EMISOR

RAZON SOCIAL

TIPO DE DOCUMENTO

RANGO - DESDE

RANGO - HASTA

FECHA DE AUTORIZACION

FIRMA CAF

LLAVE PRIVADA

LLAVE PUBLICA

VOLVER

¿Por qué tiene validez el TED?

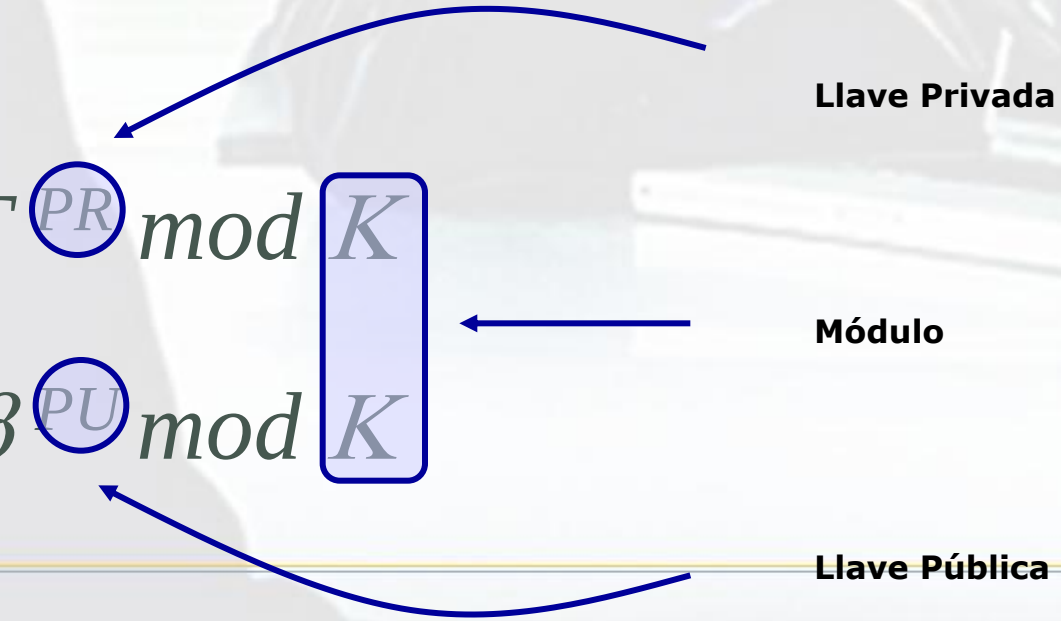
1. Permite validar que la información del TED no ha sido intervenida
2. No permite regenerarlo a partir de la información provista

FRMT – Firma del TED

ANTECEDENTES

Fundamentos de Llaves Públicas

- RSA = Algoritmo matemático permite determinar tres números que satisfacen la siguiente condición:



Ejemplo – Encriptar 123

CONSIDERANDO > PRIVADA = 17 | PUBLICA = 2753 | MODULO = 3233

ENCRIPTAR

- CYPHER = RSA(123)
- CYPHER = (123^{17}) mod 3233
- CYPHER = 337587917446653715596592958817679803 mod 3233
- CYPHER = 855

DESENCRIPTAR

- VALOR = (855^{2753}) mod 3233
- VALOR = 55565880938518988118129056142740858091687654620978905600812354234523356252836282932477692749248437 5 mod 3233
- VALOR = 123

Aspectos Importantes

- No funciona al revés, es decir, dada la llave pública no se puede aplicar el proceso inverso para reconstruir el valor cifrado

$$T^{PU} \bmod K \neq C^{PU} \bmod K$$

- Se requieren operaciones matemáticas con grandes números para poder hacer las operaciones de exponente y módulo

$$(855^{2753}) \bmod 3233 = ?$$

Paperless
digital concepts | Chile

¿COMO SE APLICA?

RSA - Proceso de Firma

- Dado un texto (TEXTO), el cálculo de la firma comprende el siguiente algoritmo:
  - A ← SHA1(TEXTO)
  - B ← PaddingPKCS1(A)
  - C ← RSA(B, PRIVADA, MODULO)
  - D ← Base64(C)

- **IMPORTANTE:**
  - *El resultado D corresponde a la firma del hash SHA1 del texto*
  - *Si se envía el TEXTO + D + PUBLICA + MODULO, receptor puede validar aplicando el proceso inverso*

RSA - Proceso de Validación

- Usuario recibe TEXTO + FIRMA + PUBLICA + MODULO.

- Proceso de validación comprende:
  - **CALCULO**
    - *A ← SHA1(TEXTO)*
  - **COMPARACION**
    - *B ← UNBASE64(FIRMA)*
    - *C ← RSA(B, PUBLICA, MODULO)*
    - *D ← SINPADDINGPKCS1(C)*
    - *¿A = D?*

RSA - Ejemplo

A ← SHA1(TEXTO)

```
A = ®™|ä'wØ□x€"Q?p+'±3¨]
```

20 BYTES

B ← PaddingPKCS1(A)

```
B = □ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ 0!0  □□+□□□□□ □□®™|ä'wØ□x€"Q?p+'±3¨]
```

64 BYTES

C ← RSA(B, PRIVADA, MODULO)

```
C = □Ÿ-□□¨`«m□`¯%c□)÷uÇÏ□Ã]□□ÅG "®®µMË□{¬±Œ·ýqw□□°à//ï¥!œÿ;<□þ□□Óî
```

64 BYTES

D ← Base64(C)

```
D = DwqfLRGBqGCrbRBgr4ljFSn3dcfPC8MKXQEDxUeglK6utU3Lf3ussYy3/XF3BAG64(
    PAv+CBTT7g==
```

88 BYTES

Paperless
digital concepts | Chile

¿DONDE ESTAN LAS LLAVES?

Las Llaves del CAF

```
<?xml version="1.0" ?>
- <AUTORIZACION>
  - <CAF version="1.0">
    - <DA>
        <RE>96915310-6</RE>
        <RS>E-PARTNERS S A</RS>
        <TD>110</TD>
      - <RNG>
          <D>1</D>
          <H>100</H>
        </RNG>
        <FA>2008-01-15</FA>
      - <RSAPK>
          <M>2lVT8m0M/vuYsrJFuM+cxXc77Wh7n54Mglc7n4/A1zXL7tn80FT5Nkt5y
          <E>Aw==</E>
        </RSAPK>
        <IDK>100</IDK>
      </DA>
      <FRMA
        algoritmo="SHA1withRSA">sYU2w5Hn9S0KC+/SWgxNprxwAM/iW9TxBNSa3U
    </CAF>
    <RSASK>-----BEGIN RSA PRIVATE KEY----- MIIBOgIBAAJBANpVU/JtDP77mLKyRt
      +TZLecvfJmTWbxBSe6VvsnlSJnvBEuYU+LsCAQMCQQCRjjf2813/UmXMdtkl373
      +ifzmv0VFAhW5NJqX9XkzUyigc4eBwrIkfIwQo/+p8OWfpBhyZkXNq9gaSWEhH
      AiEA8lq9zZZnS199+K4ow4PqsaqHPK40WIttWfgWiZjwHUcCIQDmoFl6DOMdqfK
      1VKK4vAYBA1mZIz0SxLFcwzRBl4y7QIhAKGR095kRNzqU/sexdetRyEcWih0IuW
      SOalZFu7Sr4vAiEAmcA7prNCE8ahuTjhseygEAKzmZhd+DIMg6IIi1mUIfMCIDHE
      PRIVATE KEY-----</RSASK>
    <RSAPUBK>-----BEGIN PUBLIC KEY----- MFowDQYJKoZIhvcNAQEBBQADSQAwRg
      DIJXO5+PwNc1y+7Z/NBU+TZLecvfJmTWbxBSe6VvsnlSJnvBEuYU+LsCAQM=
</AUTORIZACION>
```

SOLO SE INCORPORA ESTA SECCION EN EL TED ELIMINANDO LA LLAVE PRIVADA (RSASK) Y LA LLAVE PUBLICA (RSAPUBK)

LLAVE PRIVADA

LLAVE PUBLICA

¿Cómo leer las llaves?

```
-----BEGIN RSA PRIVATE KEY-----
MIIBOgIBAAJBALocJF/TtxJOoYdweX0+AhnMcALqVcsSmiiyL2W5JAmH4o6LxN
gTkvDYiPlRU1y53md1vrNJc9dU6KpCH7ZPNTMCAQMCQHwSwuqNJLbfFlpK+6jU
AWaISqycOTIMZsXMH5kmGAZZcqsXSeFdhbY7qXSgnChP7A5ABMETADS/llCzcm
YsrvsCIQDwNq3281T5DuiXfmeppkjfpSTSAUKRsWCH6uHzAHg75wIhAMZXOt8S
slFQlnNL+L95+Rys8OWQamHW1P9GUwEck/LVAiEAoCRz+feN+19FulRFG8QwlR
jDNquBtnZAWpyWogBQJ+8CIQCEOic/Ycw2NbmiMqXU+/toc0tDtZxBOeNU2Yyr
aGKh4wIhANI1qNouzu5cAKn3vqgAWsXRy6+H9aEX/k1EfC3ik2Dv
-----END RSA PRIVATE KEY-----
```

- Formato PEM (Privacy-Enhanced Mail)
- Permite el transporte seguro de las llaves
- Utiliza el estándar ASN.1 (Abstract Syntax Notation 1)

TED - Timbre Electrónico del DTE

NODO - TED

RE - RUT EMISOR

TD - TIPO DE DOCUMENTO

F - FOLIO

FE - FECHA EMISION

RR – RUT RECEPTOR

RSR – RAZON SOCIAL RECEPTOR

MNT – MONTO NETO

IT1 – GLOSA DETALLE 1

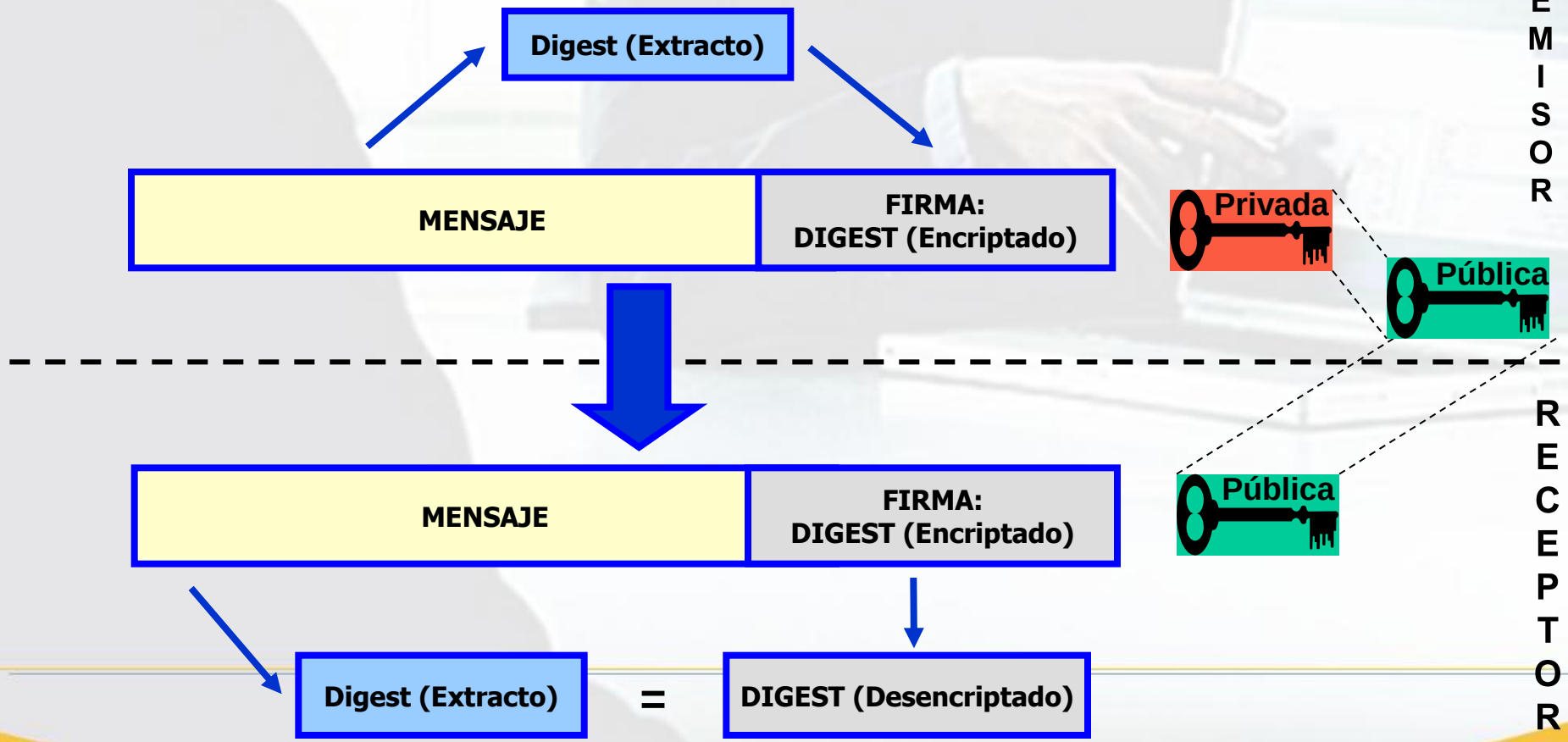
CAF

TSTED – TimeStamp TED

FRMT – Firma TED

VOLVER

Firma Electrónica (1)



Si:Firma Válida No: Firma No Válida

VOLVER

Firma Electrónica (2)

ALGORITMO

DIGEST

FIRMA

CERTIFICADO

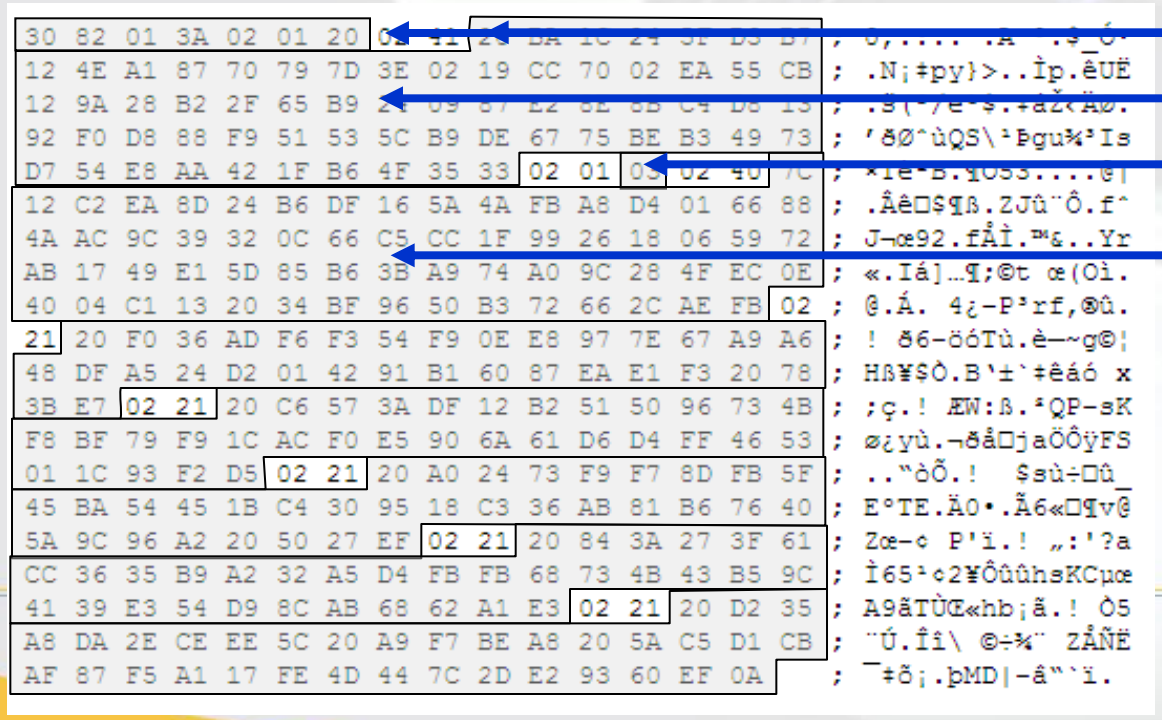
VOLVER

ASN.1 – Abstract Syntax Notation

- Norma que establece cómo representar en modo binario objetos.
- Preeliminar al XML
- Objetos Disponibles:
  - *BOOLEAN*
  - *INTEGER*
  - *BITSTRING*
  - *OCTET*
  - *:*
  - *:*

Información del PEM

- ASN.1 del PEM ➔ Formato DER
  - *Modulo, Pública, Privada, P, Q, Exp P, Exp Q, Crt Coeff*



PROCESO FINAL

Proceso Final

Dado el texto TEXTO:

A ← SHA1(TEXTO)

B ← PaddingPKCS1(A)

C ← RSA(B, PRIVADA, MODULO)

D ← Base64(C)

<CAF>

MANIPULADOS COMO NUMEROS GRANDES