12 - Digital Signatures

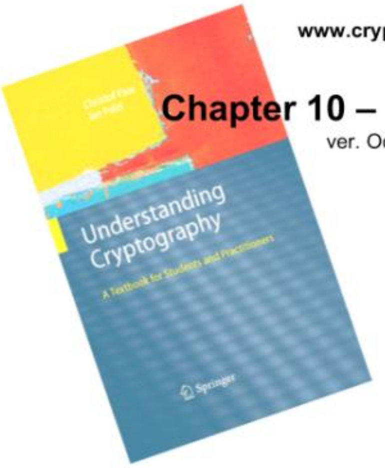
2024年10月8日 9:32

12 - Digital Signatures

Understanding Cryptography – A Textbook for Students and Practitioners

by Christof Paar and Jan Pelzl

www.crypto-textbook.com



Chapter 10 – Digital Signatures

ver. October 29, 2009

These slides were prepared by Georg Becker, Christof Paar and Jan Pelzl

Some legal stuff (sorry): Terms of Use

- The slides can used free of charge. All copyrights for the slides remain with Christof Paar and Jan Pelzl.
- The title of the accompanying book "Understanding Cryptography" by Springer and the author's names must remain on each slide.
- If the slides are modified, appropriate credits to the book authors and the book title must remain within the slides.
- It is not permitted to reproduce parts or all of the slides in printed form whatsoever without written consent by the authors.

Content of this Chapter

- The principle of digital signatures
- The RSA digital signature scheme
- The Digital Signature Algorithm (DSA)

Content of this Chapter

- **The principle of digital signatures**
- The RSA digital signature scheme
- The Digital Signature Algorithm (DSA)

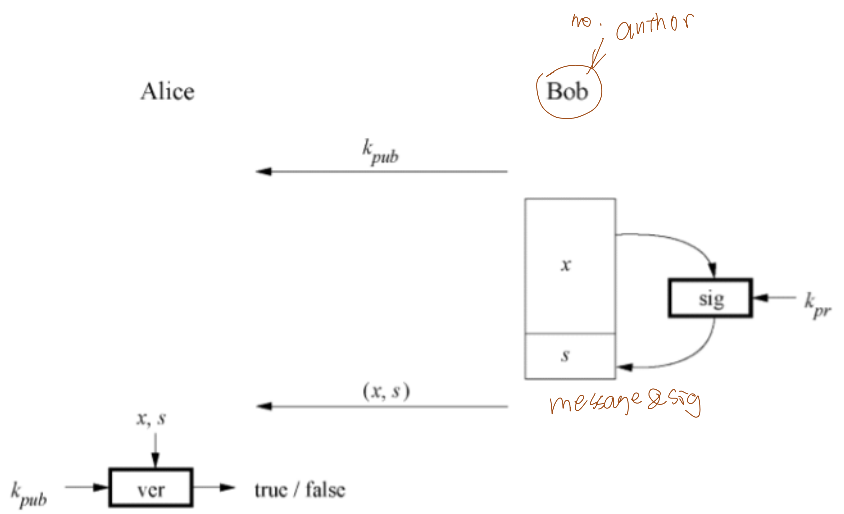
Motivation

- Alice orders a pink car from the car salesmen Bob
- After seeing the pink car, Alice states that she has never ordered it:
- How can Bob prove towards a judge that Alice has ordered a pink car? (And that he did not fabricate the order himself)

⇒ Symmetric cryptography fails because both Alice and Bob can be malicious

⇒ Can be achieved with public-key cryptography

Basic Principle of Digital Signatures



验证函数会接受三个输入

Main idea

- For a given message *x*, a digital signature is appended to the message (just like a conventional signature).
- Only the person with the private key should be able to generate the signature.
- The signature must change for every document.

⇒The signature is realized as a function with the message *x* and the private key as input.

只有拥有私钥的人才能生成有效的签名

⇒The public key and the message *x* are the inputs to the verification function.

Chapter 10 of *Understanding Cryptography* by Christof Paar and Jan Pelzl

Security Services

Digital signatures provide the following security services:

1. **Integrity**: Ensures that a message has not been modified in transit.
2. **Message Authentication:** Ensures that the sender of a message is authentic. An alternative term is data origin authentication.
3. **Non-repudiation:** Ensures that the sender of a message can not deny the creation of the message. (c.f. order of a pink car)

Confidentiality lack数字签名 **缺乏** 机密性保护功能

- 机密性意味着信息在传输过程中不被未经授权的第三方读取，而数字签名的主要功能是验证完整性、认证和不可否认性，而不是加密内容。
- 因此，若需要机密性，则需要结合 **对称加密或非对称加密** 方法来实现。

Chapter 10 of *Understanding Cryptography* by Christof Paar and Jan Pelzl

Content of this Chapter

- The principle of digital signatures
- **The RSA digital signature scheme**
- The Digital Signature Algorithm (DSA)

Main idea of the RSA signature scheme

To generate the private and public key:

- Use the same key generation as RSA encryption.

To generate the signature:

- "encrypt" the message x with the private key

$$\boxed{s = sig_{K_{priv}}(x) = x^d \bmod n}$$

- Append s to message x

To verify the signature:

- "decrypt" the signature with the public key

$$\boxed{x' = ver_{K_{pub}}(s) = s^e \bmod n}$$

- If $x=x'$, the signature is valid

- **签名生成**：
  - 签名的过程实际上是对消息 xxx 进行“加密”，但不同于通常的加密操作，这里使用的是 **私钥** ddd，而不是公钥 eee。这确保了只有拥有私钥的发送方才能生成合法的签名。
- **签名验证**：
  - 验证的过程相当于“解密”签名，但使用 **公钥** eee，这意味着任何人都可以验证签名。这提供了不可否认性，因为只有私钥持有者能够生成合法的签名。
- **安全性**：
  - RSA 签名方案的安全性依赖于 RSA **难题** 和 **整数因子分解难题** 的计算复杂度，即在已知 nnn 和 eee 的情况下，推导出 d是非常困难的。

The RSA Signature Protocol

Alice   Bob

$K_{pr} = d$
$K_{pub} = (n, e)$

$$\xleftarrow{\quad K_{pub} \quad}$$

Compute signature:
$s = sig_{k_{pr}}(x) \equiv x^d \bmod n$

$$\xleftarrow{\quad (x,s) \quad}$$

Verify signature:
$x' \equiv s^e \bmod n$
If $x' \equiv x \bmod n \rightarrow$ valid signature
If $x' \not\equiv x \bmod n \rightarrow$ invalid signature

1. 签名协议流程

- **Alice** 和 **Bob** 之间的通信。
- **Bob (签名者)**：
  - 生成公钥 $K_{pub} = (n, e)$ 和私钥 $K_{pr} = d$。
  - 计算签名 s：
    $$s = \mathrm{sig}_{K_{pr}}(x) = x^d \mod n$$
  - 将签名 s 和消息 x 一起发送给 Alice，即发送 (x, s)。
- **Alice (验证者)**：
  - 接收到消息 (x, s) 后，使用公钥 $K_{pub} = (n, e)$ 验证签名：
    $$x' = s^e \mod n$$
  - 如果 $x' = x$，则签名有效。
  - 如果 $x' \neq x$，则签名无效。

Security and Performance of the RSA Signature Scheme

Security:

The same constrains as RSA encryption: n needs to be at least 1024 bits to provide a security level of 80 bit.

⇒ The signature, consisting of s, needs to be at least 1024 bits long

Performance:

The signing process is an exponentiation with the private key and the verification process an exponentiation with the public key e.

small & light weight.

⇒ Signature verification is very efficient as a small number can be chosen for the public key.

d inverse of e mod n

↓   density & long

high

性能：

- **签名过程**：使用私钥 d 进行模幂运算（即 $x^d \mod n$）。
- **验证过程**：使用公钥 e 进行模幂运算（即 $s^e \mod n$）。
- 通常选择较小的 e（如 $e = 3$ 或 $e = 65537$），以加快验证速度。因此，**签名验证通常比签名生成更高效。**

d 的逆元：通过 $e \times d \equiv 1 \mod \phi(n)$ 计算私钥 d。

小 e：选择较小的 e 可以优化验证的计算效率，使验证过程更加轻量。

伪造

Existential Forgery Attack against RSA Digital Signature

Alice | Oscar | Bob

Bob: $K_{pr} = d$, $K_{pub} = (n, e)$

Bob → Oscar: (n,e)

Oscar → Alice: (n,e)

Oscar:

1. Choose signature:
   $s \in Z_n$
2. Compute message:
   $x \equiv s^e \bmod n$

Oscar → Alice: (x,s)

Verification:

- compute $x' \equiv s^e \bmod n$
- Compare x and x':
  $x \equiv s^e \equiv x' \bmod n$
  Signature is valid!

Chapter 10 of *Understanding Cryptography* by Christof Paar and Jan Pelzl

Existential Forgery and Padding

- An attacker can generate valid message-signature pairs (x,s)
- But an attack can only choose the signature s and NOT the message x

⇒ Attacker cannot generate messages like "Transfer $1000 into Oscar's account"

Formatting the message x according to a *padding scheme* can be used to make sure that an attacker cannot generate valid (x,s) pairs.

(A messages x generated by an attacker during an Existential Forgery Attack will not coincide with the padding scheme. For more details see Chapter 10 in *Understanding Cryptography.)*

Chapter 10 of *Understanding Cryptography* by Christof Paar and Jan Pelzl

Content of this Chapter

- The principle of digital signatures
- The RSA digital signature scheme
- **The Digital Signature Algorithm (DSA)**

Facts about the Digital Signature Algorithm (DSA)

k: 7.15.

elective curve.

"elliptic curve"（椭圆曲线）：

- 这意味着，未来可能会更多采用 **椭圆曲线数字签名算法**（ECDSA），因为它在提供相同安全级别的前提下，**密钥更短**，效率更高。

- Federal US Government standard for digital signatures (DSS)
- Proposed by the National Institute of Standards and Technology (NIST)
- DSA is based on the Elgamal signature scheme
- Signature is only 320 bits long
- Signature verification is slower compared to RSA

DSA 的特点

- **签名长度**：DSA 生成的签名长度仅为 320 **位**，相比其他签名方案更短，因此在某些应用场景下更为高效。
- **签名验证速度**：
  - DSA 的签名验证过程 **比 RSA 更慢**，这使得 DSA 更适合生成签名，而非频繁验证签名的场景。

3. 安全性与效率

- DSA 的设计初衷是提供一种安全且标准化的数字签名方案，确保数据的完整性、认证以及不可否认性。
- **验证较慢** 的原因是其基于 **离散对数问题**（DLP），需要更多的计算资源。

The Digital Signature Algorithm (DSA)

Key generation of DSA:

1. Generate a prime p with $2^{1023} < p < 2^{1024}$ 1023~1024 bits. long.
2. Find a prime divisor q of $p\text{-}1$ with $2^{159} < q < 2^{160}$
3. Find an integer α with $ord(\alpha)=q$ how many power q go there.
4. Choose a random integer d with $0<d<q$
5. Compute $\beta \equiv \alpha^d \bmod p$

The keys are:

$k_{pub} = (p,q,\alpha,\beta)$

$k_{pr} = (d)$

$k_{pub} = (p,q,\alpha,\beta)$

$k_{pr} = (d)$

The Digital Signature Algorithm (DSA)

DSA signature generation :

Given: message x, signature s, private key d and public key (p,q,α,β) 选择一个随机的临时密钥 kE

DSA 的核心这确保了即使消息相同，生成的签名也不同。

1. Choose an integer as random ephemeral key k_E with $0<k_E<q$
2. Compute $r \equiv (\alpha^{k_E} \bmod p) \bmod q$
3. Computes $s \equiv (SHA(x)+d \cdot r)\, k_E^{-1} \bmod q$

The signature consists of (r,s)

SHA denotes the hash function SHA-1 which computes a 160-bit fingerprint of message x. (See Chapter 11 of *Understanding Cryptography* for more details)

完整性、认证性和不可否认性

The Digital Signature Algorithm (DSA)

DSA signature verification

Given: message x, signature s and public key (p,q,α,β)

1. Compute auxiliary value 辅助值 $w \equiv s^{-1} \bmod q$
2. Compute auxiliary value $u_1 \equiv w \cdot SHA(x) \bmod q$
3. Compute auxiliary value $u_2 \equiv w \cdot r \bmod q$
4. Compute $v \equiv (\alpha^{u_1} \cdot \beta^{u_2} \bmod p) \bmod q$

If $v \equiv r \bmod q \rightarrow$ signature is valid

If $v \not\equiv r \bmod q \rightarrow$ signature is invalid

Proof of DSA:

We show need to show that the signature (r,s) in fact satisfied the condition $r \equiv v \bmod q$:

$s \equiv (\mathrm{SHA}(x))+d \cdot r) \cdot k_E^{-1} \bmod q$

⇔ $k_E \equiv s^{-1}\, \mathrm{SHA}(x) + d \cdot s^{-1}\, r \bmod q$

⇔ $k_E \equiv u_1 + d \cdot u_2 \bmod q$

We can raise α to either side of the equation if we reduce modulo p:

⇔ $\alpha^{kE} \bmod p \equiv \alpha^{u1+d \cdot u2} \bmod p$

Since $\beta \equiv \alpha^d \bmod p$ we can write:

⇔ $\alpha^{kE} \bmod p \equiv \alpha^{u1} \beta^{u2} \bmod p$

We now reduce both sides of the equation modulo q:

⇔ $(\alpha^{kE} \bmod p) \bmod q \equiv (\alpha^{u1} \beta^{u2} \bmod p) \bmod q$

Since $r \equiv (\alpha^{kE} \bmod p) \bmod q$ and $v \equiv (\alpha^{u1} \beta^{u2} \bmod p) \bmod q$, this expression is identical to:

⇔ $r \equiv v$

■ Example

DSA 签名示例

Alice		Bob
		Key generation:
		1. choose p = 59 and q = 29
		2. choose α = 3
	$\xleftarrow{(p,\, q,\, \alpha,\, \beta)=(59,\, 29,\, 3,\, 4)}$	3. choose private key d = 7
		4. $\beta = \alpha^d = 3^7 \equiv 4 \bmod 59$
		Sign:
		Compute has of message H(x)=26
	$\xleftarrow{(x,(r,\, s))=(x,20,\, 5)}$	1. Choose ephermal key k_E=10
Verify:		2. $r = (3^{10} \bmod 59) \equiv 20 \bmod 29$
$w \equiv 5^{-1} \equiv 6 \bmod 29$		3. $s = ((26 + 7 \cdot 20) \cdot 3) \equiv 5 \bmod 29$
$u_1 \equiv 6 \cdot 26 \equiv 11 \bmod 29$		
$u_2 \equiv 6 \cdot 20 \equiv 4 \bmod 29$		
$v = (3^{11} \cdot 4^4 \bmod 59) \bmod 29 = 20$		
$v \equiv r \bmod 29 \rightarrow$ valid signature		

Security of DSA

To solve the discrete logarithm problem in p the powerful index calculus method can be applied. But this method cannot be applied to the discrete logarithm problem of the subgroup q. Therefore q can be smaller than p. For details see Chapter 10 and Chapter 8 of *Understanding Cryptography* .

- DSA 的安全性依赖于解决 ppp 里的离散对数问题。
- **指数运算**在大素数 ppp 下非常困难，而对 qqq 的攻击更困难，因此选择较大的 ppp 和 qqq 非常重要。

p	q	hash output (min)	security levels
1024	160	160	80
2048	224	224	112
3072	256	256	128

double → birthday

Standardized parameter bit lengths and security levels for the DSA

p 的长度越大，安全性越高。

Chapter 10 of *Understanding Cryptography* by Christof Paar and Jan Pelzl

Security of DSA

一次性会话密钥的重用风险：

- 在 DSA（数字签名算法）中，如果重复使用相同的会话密钥（ephemeral key kEk_EkE），攻击者可能会推导出签名密钥 ddd，从而破坏整个签名系统的安全性

use once.

- Reuse of ephemeral key can lead to the disclosure of the signing key.
- Exercise: prove this.

- Real-world incident:
  - Sony Playstation uses the same constant as the ephemeral key in its digital signatures.
  - This was exploited in 2010 by a hacker to obtain the signing key:
  - https://www.bbc.com/news/technology-12116051

证明思路:

- 签名 (r, s) 中:

$$r = (\alpha^{k_E} \mod p) \mod q$$

$$s = (H(x) + d \cdot r) k_E^{-1} \mod q$$

- 如果 k_E 重复使用，并且攻击者能截获两个不同消息的签名对 (r, s_1) 和 (r, s_2)，他可以通过以下公式推导出 d:

$$s_1 - s_2 = (H(x_1) - H(x_2)) k_E^{-1} \mod q$$

$$k_E = \frac{H(x_1) - H(x_2)}{s_1 - s_2} \mod q$$

- 一旦获得 k_E，攻击者可以进一步解出私钥 d:

$$d = \frac{s \cdot k_E - H(x)}{r} \mod q$$

Chapter 10 of *Understanding Cryptography* by Christof Paar and Jan Pelzl

Elliptic Curve Digital Signature Algorithm (ECDSA)

椭圆曲线密码学提供与 RSA 相同级别的安全性，但所需的密钥长度更短

- Based on Elliptic Curve Cryptography (ECC)
- Bit lengths in the range of 160-256 bits can be chosen to provide security equivalent to 1024-3072 bit RSA (80-128 bit symmetric security level)
- One signature consists of two points, hence the signature is twice the used bit length (i.e., 320-512 bits for 80-128 bit security level). 签名的长度是密钥长度的两倍。例如，对于 160 位 ECC 密钥，签名长度为 320 位。
- The shorter bit length of ECDSA often result in shorter processing time 更短的密钥长度意味着更快的处理时间和更小的存储需求，因此更适合在资源受限的环境（如移动设备和嵌入式系统）中使用。

For more details see Section 10.5 in *Understanding Cryptography*

（如移动设备和嵌入式系统）中使用。

For more details see Section 10.5 in *Understanding Cryptography*

Lessons Learned

- Digital signatures provide message integrity, message authentication and non-repudiation.

find key ?

- RSA is currently the most widely used digital signature algorithm.
- Competitors are the Digital Signature Standard (DSA) and the Elliptic Curve Digital Signature Standard (ECDSA).
- RSA verification can be done with short public keys *e*. Hence, in practice, RSA verification is usually faster than signing.
- DSA and ECDSA have shorter signatures than RSA
- In order to prevent certain attacks, RSA should be used with padding.
- The modulus of DSA and the RSA signature schemes should be at least 1024-bits long. For true long-term security, a modulus of length 3072 bits should be chosen. In contrast, ECDSA achieves the same security levels with bit lengths in the range 160–256 bits.

ECDSA 可以在 160-256 **位**的密钥长度下达到与 1024-3072 **位** RSA 相同的安全水平。因此，
ECDSA 更适合在资源受限的环境中使用，如嵌入式系统和移动设备。