GMR: A strong cryptographic signature system

Software Security

Steffen Helke
Chair of Software Engineering

5th December 2018

Brandenburgische
Technische Universität
Cottbus - Senftenberg

Objectives of today's lecture

- ➜ Getting to know *how to perform a square test* for a given number k when the prime factors of n are known
- ➜ Understanding the basic idea of *collision-resistant permutations* in the context of a signature process
- ➜ Being able to apply the *asymmetric signature system GMR* to a small example

Prime Factorization

Definition

The prime factorization of a natural number n is the product

$$n = p_1^{e_1} \cdot p_2^{e_2} \cdot \quad \dots \quad \cdot p_k^{e_k}$$

where $p_1, \dots, p_k$ are different prime numbers in pairs and the exponents are positive natural numbers, i.e. $e_1, \dots, e_k \in \mathbb{N}^+$

Algorithms

- Pollard's rho algorithm
- Quadratic sieve algorithm
- Number field sieve

➜ No polynomial algorithm for prime factor decomposition has been found yet!

Fundamentals of Number Theory

Factorization is hard

There is no polynomial algorithm to efficiently calculate the prime numbers p and q from a given n, so that $p \cdot q = n$ applies

Implications

There are two other algorithms that are as hard as factorization

1. Calculating a square root $\bmod\ n$
2. Testing for a square $\bmod\ n$[1]

➜ However, if you know p and q, then both tasks can be solved efficiently, e.g. root extraction using the CRA (Chinese Remainder Algorithm)!

[1] Also called the *quadratic residuacity problem*

What are squares in number theory?

Quadratic Residues

A number a is a quadratic residue in $(\bmod\ n)$, iff

$x^2 \equiv a \bmod n$

Set of all Quadratic Residues for a given n

$QR_n = \{x : \mathbb{Z}_n^* \mid \exists\, y : \mathbb{Z}_n^* \bullet y^2 \equiv x \bmod n\}$, where

$\mathbb{Z}_n = \{0, \ldots, n-1\}$

$\mathbb{Z}_n^* = \{a : \mathbb{Z}_n \mid gcd(a, n) = 1\}$

Each $x \in QR_n$ has an unique root in QR_n

$W_x = \{w : \mathbb{Z}_n \mid w^2 \equiv x \bmod n\}$, i.e. $\mid W_x \cap QR_n \mid = 1$

How to calculate a square test efficiently?

- For the prime numbers p and q with $n = p \cdot q$ is valid:

  $x \in QR_n \Leftrightarrow x \in QR_p \wedge x \in QR_q$

- Legendre-Symbol (in generalized form Jacobi-Symbol) can used for a square test of a given x, only valid if p is prime number

  $$\left(\frac{x}{p}\right) = \begin{cases} 1 & \text{if } x \text{ is a quadratic residue } (\bmod\ p) \\ -1 & \text{if } x \text{ is a quadratic nonresidue } (\bmod\ p) \\ 0 & \text{if } x \equiv 0 \bmod p \end{cases}$$

- Jacobi-Test with p and q as prime numbers and $n = p \cdot q$

  $\left(\frac{a}{n}\right) = \left(\frac{a}{p}\right) \cdot \left(\frac{a}{q}\right)$

  $\boxed{\text{It is only valid: } a \in QR_n \Rightarrow \left(\frac{a}{n}\right) = 1}$

- Euler's criterion defines for an odd prime number p

  $\left(\frac{a}{p}\right) = a^{\frac{p-1}{2}}$

Example: How to calculate a square test?

The following prime numbers are given

- $p = 3$ and $q = 7$ with $n = p \cdot q = 21$

We use $s = 1$ with $s \in \mathbb{Z}_{21}$ to calculate a square test

1. Jacobi-Test for $p = 3$: $(\left(\frac{1}{3}\right) = 1^{\frac{3-1}{2}} \equiv 1 \bmod 3) \Rightarrow 1 \in QR_3$
2. Jacobi-Test for $p = 7$: $(\left(\frac{1}{7}\right) = 1^{\frac{7-1}{2}} \equiv 1 \bmod 7) \Rightarrow 1 \in QR_7$
3. Conclusion: 1 is square, because $1 \in QR_3 \wedge 1 \in QR_7 \Rightarrow 1 \in QR_{21}$

There exists 4 roots for the square $s = 1$

- $f(1) = f(8) = f(13) = f(20) = 1$ with $f(x) = x^2 \bmod 21$

Which root of a square is again a square?

There exists 4 roots for the square $s = 1$

- $f(1) = f(8) = f(13) = f(20) = 1$ with $f(x) = x^2 \bmod 21$

Calculating a square test for all roots

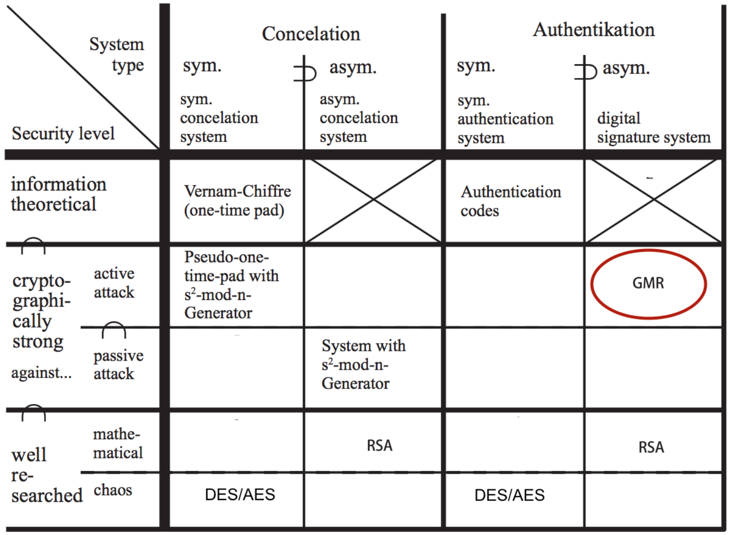
- $\left(\frac{1}{21}\right) = \left(\frac{1}{3}\right) \cdot \left(\frac{1}{7}\right) = (1 \bmod 3) \cdot (1 \bmod 7) = 1 \cdot 1 = 1$
- $\left(\frac{8}{21}\right) = \left(\frac{8}{3}\right) \cdot \left(\frac{8}{7}\right) = (8 \bmod 3) \cdot (8^3 \bmod 7) = -1 \cdot 1 = -1$
- $\left(\frac{13}{21}\right) = \left(\frac{13}{3}\right) \cdot \left(\frac{13}{7}\right) = (13 \bmod 3) \cdot (13^3 \bmod 7) = 1 \cdot -1 = -1$
- $\left(\frac{20}{21}\right) = \left(\frac{20}{3}\right) \cdot \left(\frac{20}{7}\right) = (20 \bmod 3) \cdot (20^3 \bmod 7) = -1 \cdot -1 = 1$

We obtain 2 roots with Jacobi-Symbol 1

- However only 1 is square, because $20 \notin QR_3 \wedge 20 \notin QR_7 \Rightarrow 20 \notin QR_{21}$
- Note, the negation of the square is always equal to the other root with Jacobi-Symbol 1, e.g. $-1 \equiv 20 \bmod 21$

GMR: A Strong Cryptographic Signature System

GMR: A Strong Cryptographic Signature System



System type Security level		Concelation		Authentikation	
		sym. ⊅	asym.	sym. ⊅	asym.
		sym. concelation system	asym. concelation system	sym. authentication system	digital signature system
information theoretical		Vernam-Chiffre (one-time pad)	☒	Authentication codes	☒
crypto-graphically strong against...	active attack	Pseudo-one-time-pad with s^2-mod-n-Generator			GMR
	passive attack		System with s^2-mod-n-Generator		
well researched	mathematical		RSA		RSA
	chaos	DES/AES		DES/AES	

Source: *Andreas Pfitzmann: Security in IT-Networks*, 2012

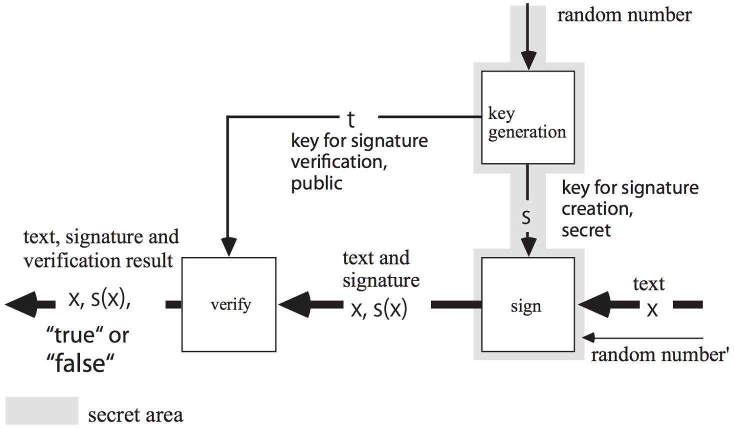
Signature System GMR

- Inventors
  - Shafi **G**oldwasser
  - Silvio **M**icali
  - Ronald L. **R**ivest
- First conceptual presentation **1984**
- Revised and improved version published **1988**
- First efficient implementation **1991**
- Cryptographically strong signature system,
  GMR is stronger than RSA!

➜ But unfortunately not used today!

Asymmetric Digital Signature System

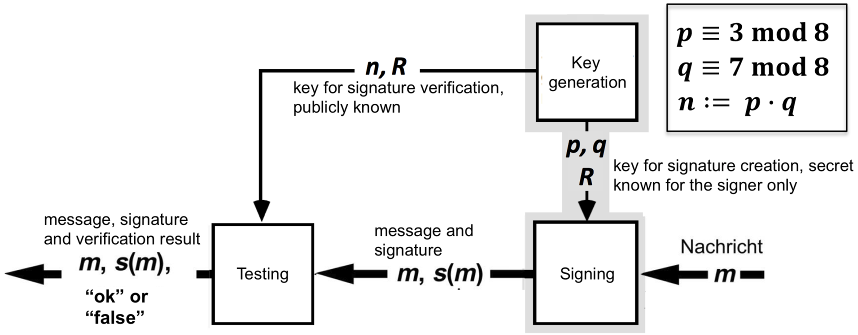
- Anyone is able to test the signature with *t* (public key)
- The key *s* is only known by the signer (secret key)


random number
key generation
t
key for signature verification, public
s
key for signature creation, secret
text, signature and verification result
x, s(x), “true“ or “false“
verify
text and signature
x, s(x)
sign
text
x
random number'
secret area

Source: *Andreas Pfitzmann: Security in IT-Networks*, 2012

Functioning of the Signature System GMR

- Generating the signature $s(m)$ with (p, q) and the reference R
- Testing $s(m)$ with n and the reference R

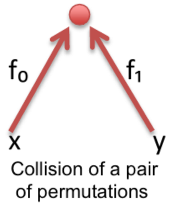


Source: *Andreas Pfitzmann: Security in IT-Networks*, 2012

Fundamentals of GMR

Conceptual Idea

- Constructing signatures with a *collision resistant permutation* pair (f_0, f_1)
- *Collisions are hard to find* under a cryptographic assumption, e.g. under the factorization assumption
- Collisions exist, however, because the permutations have identical domains



Collisions

- If $f_0(x) = f_1(y)$ and $x \neq y$ are valid, f_0 has a collision with f_1
- For GMR, a fix permutation pair (f_0, f_1) is given

Functions (Permutations) for GMR

Permutation

A permutation $f : X \to X$ is a bijective mapping, where the domain X is a countable set

Use of two square functions as permutations

$$f_0(x) = \begin{cases} x^2 \bmod n & \text{if } (x^2 \bmod n) < \frac{n}{2} \\ -x^2 \bmod n & \text{else} \end{cases}$$

$$f_1(x) = \begin{cases} 4 * x^2 \bmod n & \text{if } (4 * x^2 \bmod n) < \frac{n}{2} \\ -4 * x^2 \bmod n & \text{else} \end{cases}$$

Domain for f_0 and f_1

$$D_n = \{x : \mathbb{Z}_n^* \mid \left(\frac{x}{n}\right) = 1, x < \frac{n}{2}\}$$

How to generate a signature using GMR?

Example for a signature

- Suppose that a sender wants to generate a signature of the message $m = 110$ and a reference R has been defined
- Than the signature can be generated as follows

$$s(m) = s(110) = f_0^{-1}(f_1^{-1}(f_1^{-1}(R)))$$

Testing a signature

- The signature can be tested by the receiver with the public reference R as follows

$$R = f_1(f_1(f_0(s(110))))$$

How to construct permutations for GMR?

Requirements

- Both squaring functions f_0 and f_1 should be permutations, that can also be reversed
- Inverse function is extracting the root, hard to calculate without prime numbers p and q

  ➜ *good for signature creation*

Problem

- Not every number is a square number, e.g. -1
- Each square number has exactly 4 roots ($\bmod\ n$) with $n = p \cdot q$ ➜ *ordinary squaring is no permutation*

Solution

➜ Restrict the domain D of f_0 and f_1 and select one of 4 roots

Example: Squaring is a non-injective function

The following prime numbers are given

- $p = 3$ and $q = 7$ with $n = p \cdot q = 21$

$$f_0(x) = \begin{cases} x^2 \bmod n & \text{if } (x^2 \bmod n) < \frac{n}{2} \\ -x^2 \bmod n & \text{else} \end{cases}$$

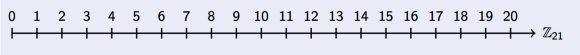
The inverse function f_0^{-1} is ambiguous

- $f_0(1) = f_0(8) = f_0(13) = f_0(20) = 1$
- $f_0(2) = f_0(5) = f_0(16) = f_0(19) = 4$
- $f_0(4) = f_0(10) = f_0(11) = f_0(17) = 5$

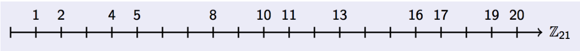
➜ We need to restrict the definition range to obtain a bijective function (permutations)

Example for a domain restriction $n = p \cdot q = 3 \cdot 7 = 21$

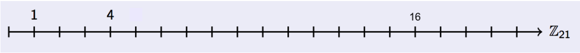
1. Start with residue class ring of n, such that $D = \mathbb{Z}_{12} = \{0, \ldots, 20\}$



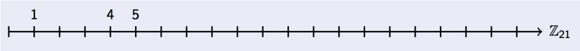
2. Select elements that have an inverse $D = \mathbb{Z}_{21}^* = \{x : \mathbb{Z}_{21} \mid \gcd(x, 21) = 1\}$



3. Select all quadratic residues $D = \{x : \mathbb{Z}_{21}^* \mid x \in QR_{21}\}$



4. Replace each quadratic residue s with $s \geq \frac{n}{2}$ by a representative r with $r \equiv -s \bmod n$[1], e.g. 5 is representative for 16 with $-16 \equiv 5 \bmod 21$ $D = \{x : \mathbb{Z}_{21}^* \mid \left(\frac{x}{21}\right) = 1, x < \frac{21}{2}\}$



[1] There are two reasons for this procedure: First we obtain using the representative a smaller signature, which is good for the performance of the signature system. Second for the test function it makes no difference if the original square or just it's representative is used. However, if you like to calculate square roots from a representative you must transform it back to the original square. Hence Step 4 is probably not useful for intermediate results of a signature creation, only for the last step.

Example for a domain restriction $n = p \cdot q = 3 \cdot 7 = 21$

➜ For the test function, it is irrelevant whether the original square or only its representative is used

$$\begin{aligned} f_0(5) &= 5^2 &&\equiv 25 &&\equiv 4 \bmod 21 \\ f_0(16) &= 16^2 &&\equiv 252 &&\equiv 4 \bmod 21 \\ f_1(5) &= -4 \cdot 5^2 &&\equiv -100 &&\equiv 5 \bmod 21 \\ f_1(16) &= -4 \cdot 16^2 &&\equiv -1024 &&\equiv 5 \bmod 21 \end{aligned}$$

Why works the Jacobi-Test also for the representative?

- Domain is restricted by $D_{21} = \{x : \mathbb{Z}_{21}^* \mid \left(\frac{x}{21}\right) = 1, x < \frac{21}{2}\} = \{1, 4, 5\}$
- Note, only the representative (the negation of the square) and the square have the Jacobi-value 1, e.g. $\left(\frac{5}{21}\right) = \left(\frac{16}{21}\right) = 1$, but other elements not, e.g. the other roots of 4 have $\left(\frac{2}{21}\right) = \left(\frac{19}{21}\right) = -1$
- Furthermore, either the square or the representative is lower than $\frac{n}{2}$, but not both, e.g. $5 < \frac{21}{2}$ and $16 \geq \frac{21}{2}$

Inverse Functions (Permutations) for GMR

Two square functions for testing a signature

$$f_0(x) = \begin{cases} x^2 \bmod n & \text{if } (x^2 \bmod n) < \frac{n}{2} \\ -x^2 \bmod n & \text{else} \end{cases}$$

$$f_1(x) = \begin{cases} 4 * x^2 \bmod n & \text{if } (4 * x^2 \bmod n) < \frac{n}{2} \\ -4 * x^2 \bmod n & \text{else} \end{cases}$$

Two root extraction functions for signing a message

$$f_0^{-1}(x) = \begin{cases} \sqrt{y} \bmod n^{1} & \text{if } (\sqrt{y} \bmod n) < \frac{n}{2} \\ -\sqrt{y} \bmod n & \text{else} \\ & \text{whereby if } x \in QR_n \text{ then } y = x \text{ else } y = \text{-}\, x \bmod n \end{cases}$$

$$f_1^{-1}(x) = \begin{cases} \sqrt{\frac{y}{4}} \bmod n & \text{if } (\sqrt{\frac{y}{4}} \bmod n) < \frac{n}{2} \\ -\sqrt{\frac{y}{4}} \bmod n & \text{else} \\ & \text{whereby if } \frac{x}{4} \in QR_n \text{ then } y = x \text{ else } y = \text{-}\, x \bmod n \end{cases}$$

[1] Note $\sqrt{y}$ yields a unique value because we consider only the root, which is again an quadratic residue

Example for GMR

How to create a signature of a message?

- We assume $p = 3$ and $q = 7$ with $n = p \cdot q = 21$
- Calculate $f_1^{-1}(f_0^{-1}(R))$ for the reference $R = 5$ with $R \in D_{21}$ to sign the message $m = 01$

Example: How to create a signature?

Procedure for $f_0^{-1}(5)$

1. Test, whether 5 or -5 is a square, i.e. check $5 \in QR_{21}$
2. Depending on the result in (**1.**)

   calculate roots either for $y = 5$ or for $y = -5$

   $y_3 = y^{\frac{3+1}{4}} \bmod 3$ and $y_7 = y^{\frac{7+1}{4}} \bmod 7$
3. Combine the intermediate results from (**2.**) with the CRA in such a way that you will get a square again

   $y = CRA(\pm y_3, \pm y_7, 3, 7)$
4. Test, whether the result y is within the domain of definition, e.g. $y < \frac{21}{2}$. If not, build the negation of y, e.g. $y = -y \bmod 21$

Step 1: Test, whether 5 is a square

Test for quadratic residue

$$5 \in QR_{21} \Leftrightarrow 5 \in QR_3 \wedge 5 \in QR_7$$

Jacobi-Test with Euler's criterion

- for $p = 3$ we obtain $\left(\frac{5}{3}\right) = 5^{\frac{3-1}{2}} \equiv -1 \mod 3$

  $\Rightarrow 5 \notin QR_3$
- for $p = 7$ we obtain $\left(\frac{5}{7}\right) = 5^{\frac{7-1}{2}} = 5^3 \equiv -1 \mod 7$

  $\Rightarrow 5 \notin QR_7$

➜ 5 is not a quadratic residue, i.e. $5 \notin QR_{21}$

➜ However the Jacobi-Test for $-5 \equiv 16 \mod 21$ is successful. Hence we use in the following 16 as reference for signing!

Step 2: Calculate the roots of 16, mod p and mod q

Formulas

- $y_p = y^{\frac{p+1}{4}} \bmod p$
- $y_q = y^{\frac{q+1}{4}} \bmod q$

Computing the square roots

- $y_3 = 16^{\frac{3+1}{4}} = 16^1 = 1 \mod 3$
- $y_7 = 16^{\frac{7+1}{4}} = 16^2 = 4 \mod 7$

➜ Now we have two intermediate results $y_3 = 1$ and $y_7 = 4$

Note

➜ The calculation rule can only be used under the condition $p \equiv q \equiv 3 \bmod 4$!

Step 3: Combine the intermediate results with CRA

Chinese Remainder Algorithm (CRA)

$$CRA\,(y_p, y_q, p, q) = u \cdot p \cdot y_q + v \cdot q \cdot y_p \mod n$$

Instantiation

$$CRA\,(1, 4, 3, 7) = u \cdot 3 \cdot 4 + v \cdot 7 \cdot 1 \mod 21,$$

How to calculate the base vectors u and v?

- The integer variables u and v must fulfill the condition $\gcd(3, 7) = u \cdot 3 + v \cdot 7 = 1$
- Values for u and v can be calculated using the *Extended Euclidean algorithm*

Step 3: Combine the intermediate results with CRA

Extended Euclidean algorithm

$$7 = 2 \cdot 3 + 1 \qquad (q = s_1 \cdot p + r_1)$$
$$3 = 3 \cdot 1 + 0 \qquad (p = s_2 \cdot r_1 + r_2)$$

In reverse order, i.e. solve all equations to the rest and then insert them step by step

$$0 = 3 - 3 \cdot 1 \qquad \text{(skip this equation, because } r_2 = 0)$$
$$1 = 1 \cdot 7 - 2 \cdot 3 \qquad (r_1 = q - s_1 \cdot p)$$

➜ The base vectors are $u = -2$ and $v = 1$

➜ Results in $CRA\,(1, 4, 3, 7) = -2 \cdot 3 \cdot 4 + 1 \cdot 7 \cdot 1 \equiv 4 \bmod 21$

➜ **Note**: In addition, check whether the root 4 is a square again

Step 4: Test whether 4 is within the definition domain

Check the condition $4 \in D_{21}$

$$4 < \frac{21}{2}$$

Conclusions

➜ The signature of $f_0^{-1}(5) = 4$

➜ To sign the whole message m with $m = 01$ we need additionally to calculate $f_1^{-1}(4)$

Example: How to create a signature?

Procedure for $f_1^{-1}(4)$

1. Test, whether $\frac{4}{4}$ is square, i.e. check $\frac{4}{4} \in QR_{21}$, Note the division is a multiplication with the inverse of 4, i.e. $\frac{4}{4} = 4 \cdot 4^{-1} \bmod 21$
2. Depending on the result in (**1.**)

   calculate roots either for $y = \frac{4}{4}$ or for $y = \frac{-4}{4}$

   $y_3 = y^{\frac{3+1}{4}} \bmod 3$ and $y_7 = y^{\frac{7+1}{4}} \bmod 7$
3. Combine the intermediate results from (**2.**) with the CRA in such a way that you will get a square again

   $y = CRA(\pm y_3, \pm y_7, 3, 7)$
4. Test, whether the result y is within the domain of definition, e.g. $y < \frac{21}{2}$. If not, build the negation of y, e.g. $y = -y \bmod 21$

Step 1: Test, whether $\frac{4}{4}$ is a square

Test for quadratic residue

- $\frac{4}{4} \in QR_{21} \Leftrightarrow \frac{4}{4} \in QR_3 \wedge \frac{4}{4} \in QR_7$

How to calculate the multiplicative inverse of 4?

- The multiplicative inverse i has to fulfill the following condition $i \cdot 4 + n \cdot 21 = 1$
- We solve this by the *Extended Euclidean algorithm*

  $21 = 5 \cdot 4 + 1$ ➜ $1 = 1 \cdot 21 - 5 \cdot 4$ ➜ $i = 4^{-1} = -5 \equiv 16 \bmod 21$

Test using the multiplicative inverse

- $\frac{4}{4} = 4 \cdot 4^{-1} = 4 \cdot 16 \equiv 1 \bmod 21$ ➜ $1 \in QR_{21} \Leftrightarrow 1 \in QR_3 \wedge 1 \in QR_7$
- $\left(\frac{1}{3}\right) = 1^{\frac{3-1}{2}} \equiv 1 \bmod 3$ ➜ $1 \in QR_3$
- $\left(\frac{1}{7}\right) = 1^{\frac{7-1}{2}} \equiv 1 \bmod 7$ ➜ $1 \in QR_7$

Conclusion: $\frac{4}{4} \in QR_{21}$ because $1 \in QR_{21}$

Step 2: Calculate the roots of 1, mod p and mod q

Formulas

- $y_p = y^{\frac{p+1}{4}} \bmod p$
- $y_q = y^{\frac{q+1}{4}} \bmod q$

Computing the square roots

- $y_3 = 1^{\frac{3+1}{4}} = 1^1 = 1 \bmod 3$
- $y_7 = 1^{\frac{7+1}{4}} = 1^2 = 1 \bmod 7$

➜ Now we have two intermediate results $y_3 = 1$ and $y_7 = 1$

Note

➜ The calculation rule can only be used under the condition $p \equiv q \equiv 3 \bmod 4$!

Step 3 & 4: Combine the intermediate results with CRA

Chinese Remainder Algorithm (CRA)

$$CRA(y_p, y_q, p, q) = u \cdot p \cdot y_q + v \cdot q \cdot y_p \bmod n$$

Instantiation

$$CRA(1,1,3,7) = u \cdot 3 \cdot 1 + v \cdot 7 \cdot 1 \bmod 21,$$

The base vectors u and v are already known

$$CRA(1,1,3,7) = -2 \cdot 3 \cdot 1 + 1 \cdot 7 \cdot 1 \equiv 1 \bmod 21,$$

➜ Furthermore, we have already checked the following conditions

$1 \in QR_{21}$ and $1 \in D_{21}$, i.e. $1 < \frac{21}{2}$

Conclusion: $f_1^{-1}(f_0^{-1}(5)) = 1$, i.e. the signature of $m = 01$ is 1