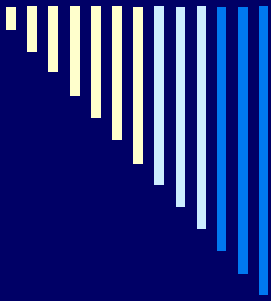



Post-Quantum Cryptography



Paulo S. L. M. Barreto

LARC/PCS/EPUSP

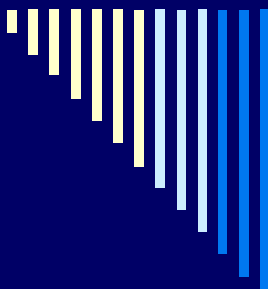


Applications of the Extended Euclidean Algorithm



GCD(A, B) with $A, B \in \mathbb{N}$

```
 $F \leftarrow A, G \leftarrow B$   
while ( $F > 0$ ) {  
    if ( $F < G$ ) {  
         $F \leftrightarrow G$   
    }  
     $h \leftarrow \lfloor F/G \rfloor$   
     $F \leftarrow F - hG$   
}  
return  $G$ 
```



GCD($f(x), g(x)$) with $f, g \in \mathbb{K}[x]$

```
 $F \leftarrow f, G \leftarrow g$   
if ( $\text{deg}(G) < 0$ ) {  
     $F \leftrightarrow G$   
}  
while ( $\text{deg}(F) \geq 0$ ) {  
     $F \leftrightarrow G$   
    while ( $\text{deg}(F) \geq \text{deg}(G)$ ) {  
         $j \leftarrow \text{deg}(F) - \text{deg}(G), h \leftarrow F_{\text{deg}(F)} / G_{\text{deg}(G)}$   
         $F \leftarrow F - h x^j G$   
    }  
}  
return ( $G \neq 0$ ) ?  $G / G_{\text{deg}(G)}$  : 0;
```



$A^{-1} \pmod{M}$ with $A, M \in \mathbb{N}$

```
// invariants:  $F = BA + XM, G = CA + YM$   
 $F \leftarrow A, B \leftarrow 1, G \leftarrow M, C \leftarrow 0$  //  $X \leftarrow 0, Y \leftarrow 1$   
while ( $F > 1$ ) {  
    if ( $F < G$ ) {  
         $F \leftrightarrow G, B \leftrightarrow C$  //  $X \leftrightarrow Y$   
    }  
     $h \leftarrow \lfloor F/G \rfloor$   
     $F \leftarrow F - hG, B \leftarrow B - hC$  //  $X \leftarrow X - hY$   
}  
if ( $F = 1$ ) return  $B$  else "not invertible"
```



$f(x)^{-1} \pmod{g(x)}$
with $f, g \in \mathbb{K}[x]$

// invariants: $F = Bf + Xg, G = Cf + Yg$

$F \leftarrow f, G \leftarrow g, B \leftarrow 1, C \leftarrow 0$

while ($\deg(F) > 0$) {

$F \leftrightarrow G, B \leftrightarrow C$

while ($\deg(F) \geq \deg(G)$) {

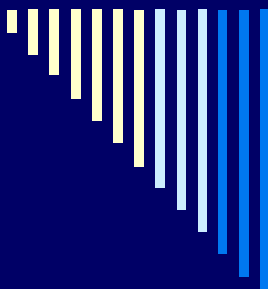
$j \leftarrow \deg(F) - \deg(G), h \leftarrow F_{\deg(F)} / G_{\deg(G)}$

$F \leftarrow F - h x^j G, B \leftarrow B - h x^j C$

}

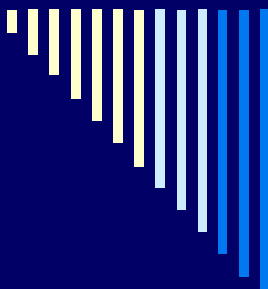
}

if ($F \neq 0$) **return** B / F_0 **else** "not invertible"



Decoding a binary Goppa syndrome $s(x)$

```
 $v(x) \leftarrow \sqrt{x + 1/s(x)} \bmod g(x)$  // extended Euclid!  
 $F \leftarrow v, G \leftarrow g, B \leftarrow 1, C \leftarrow 0, t \leftarrow \deg(g)$   
while ( $\deg(G) > \lfloor t/2 \rfloor$ ) {  
     $F \leftrightarrow G, B \leftrightarrow C$   
    while ( $\deg(F) \geq \deg(G)$ ) {  
         $j \leftarrow \deg(F) - \deg(G), h \leftarrow F_{\deg(F)} / G_{\deg(G)}$   
         $F \leftarrow F - h x^j G, B \leftarrow B - h x^j C$   
    }  
}  
}  
 $\sigma(x) \leftarrow G(x)^2 + xC(x)^2$   
return  $\sigma$  // error locator polynomial
```



Decoding an alternant syndrome $s(x)$

$F \leftarrow s, G \leftarrow x^r, B \leftarrow 1, C \leftarrow 0$ // N.B. usually $r = 2t$

while ($\deg(G) \geq \lfloor r/2 \rfloor$) {

$F \leftrightarrow G, B \leftrightarrow C$

while ($\deg(F) \geq \deg(G)$) {

$j \leftarrow \deg(F) - \deg(G), h \leftarrow F_{\deg(F)} / G_{\deg(G)}$

$F \leftarrow F - h x^j G, B \leftarrow B - h x^j C$

}

}

$\sigma(x) \leftarrow C(x) / C_0$ // error locator polynomial, $\sigma(0) = 1$

$\omega(x) \leftarrow G(x) / C_0$ // error evaluator polynomial

return σ, ω // for $\leq \lfloor r/2 \rfloor$ errors