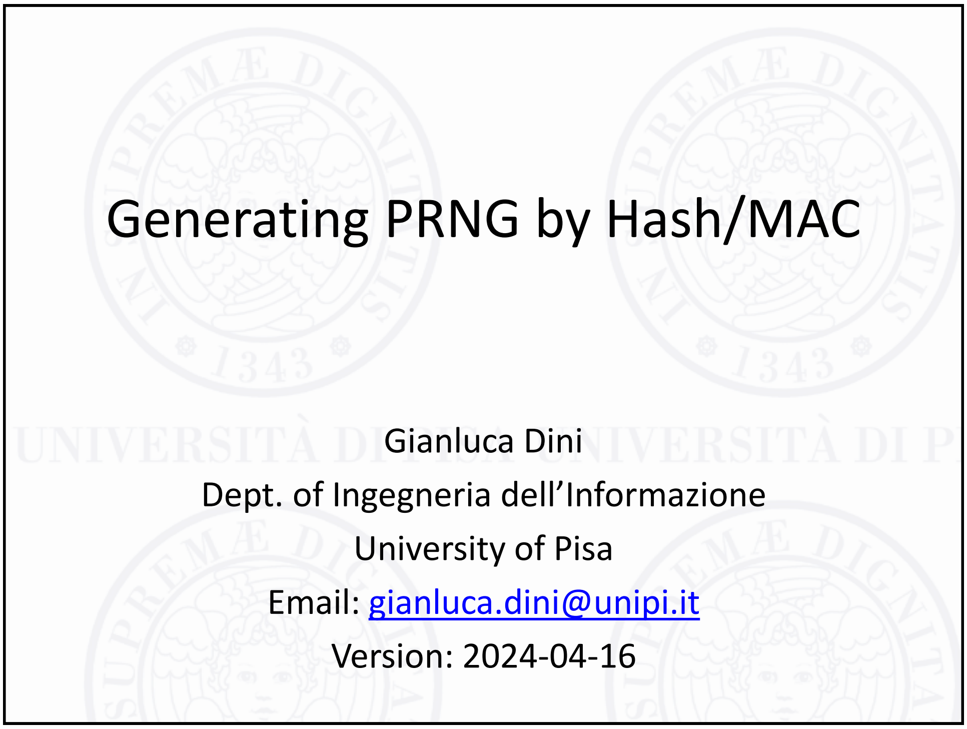
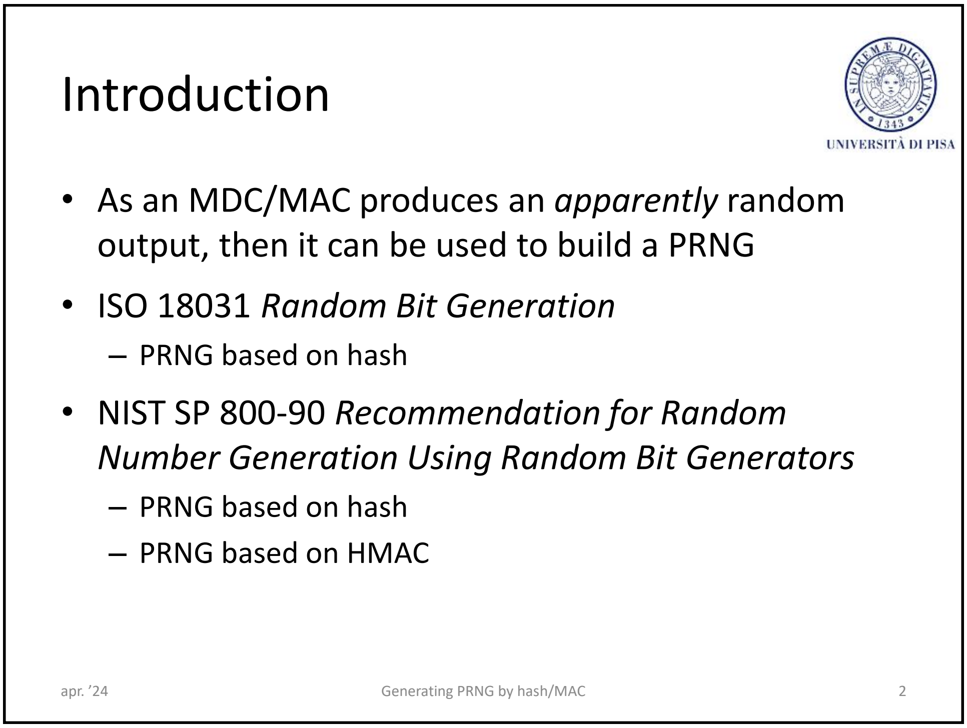


Generating PRNG by Hash/MAC

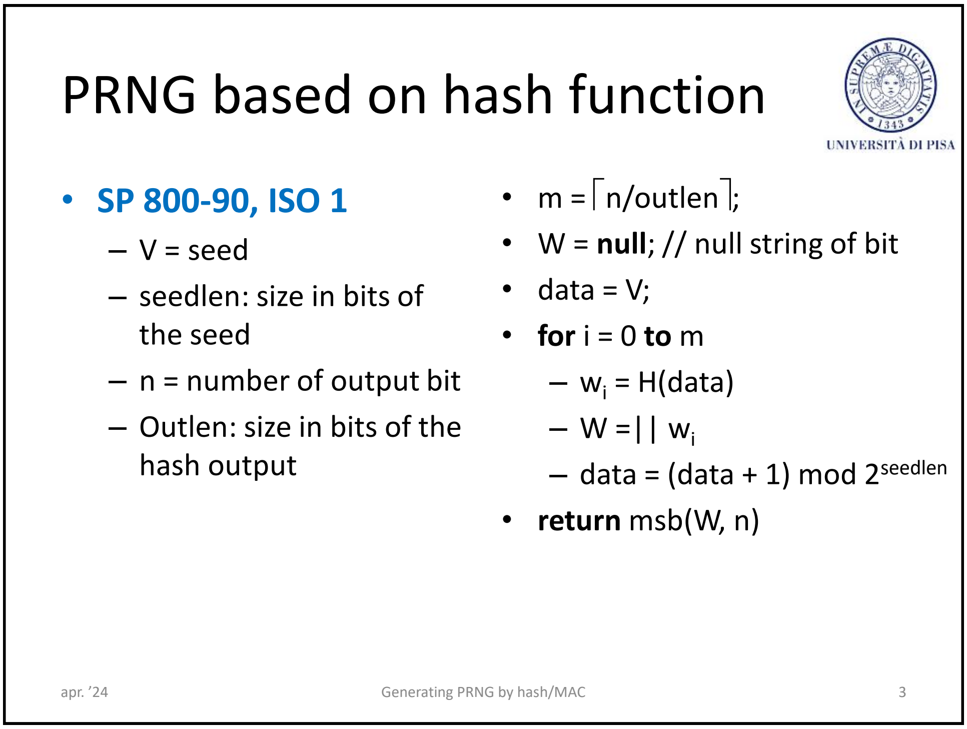
Gianluca Dini
Dept. of Ingegneria dell’Informazione
University of Pisa
Email: gianluca.dini@unipi.it

Version: 2024-04-16



Introduction

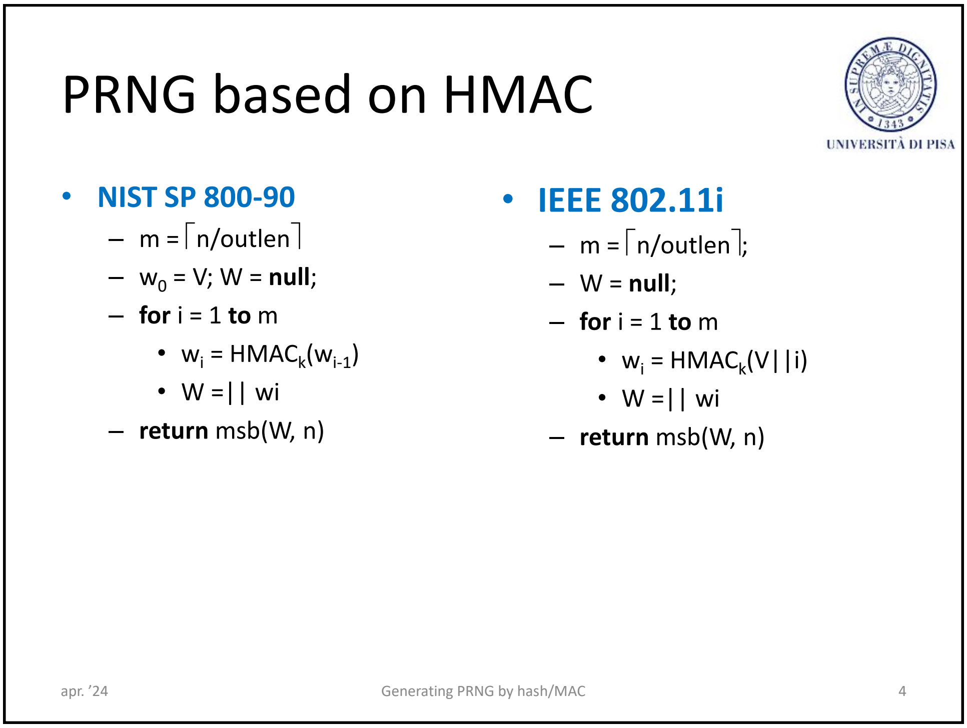
- As an MDC/MAC produces an *apparently* random output, then it can be used to build a PRNG
- ISO 18031 *Random Bit Generation*
  - PRNG based on hash
- NIST SP 800-90 *Recommendation for Random Number Generation Using Random Bit Generators*
  - PRNG based on hash
  - PRNG based on HMAC



PRNG based on hash function

- **SP 800-90, ISO 1**
  - V = seed
  - seedlen: size in bits of the seed
  - n = number of output bit
  - Outlen: size in bits of the hash output

```
• m = ⌈n/outlen⌉;
• W = null; // null string of bit
• data = V;
• for i = 0 to m
    – $w_i$ = H(data)
    – W =|| $w_i$
    – data = (data + 1) mod $2^{seedlen}$
• return msb(W, n)
```



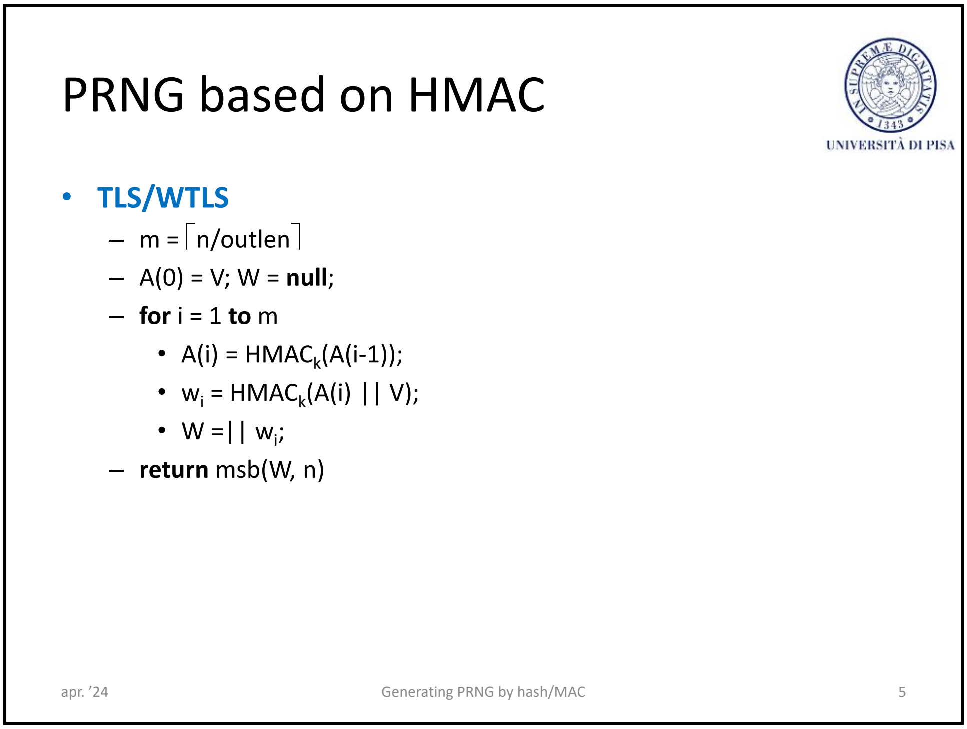
PRNG based on HMAC

- **NIST SP 800-90**

```
– m = ⌈n/outlen⌉
– $w_0$ = V; W = null;
– for i = 1 to m
    • $w_i$ = $HMAC_k(w_{i-1})$
    • W =|| wi
– return msb(W, n)
```

- **IEEE 802.11i**

```
– m = ⌈n/outlen⌉;
– W = null;
– for i = 1 to m
    • $w_i$ = $HMAC_k$(V||i)
    • W =|| wi
– return msb(W, n)
```



PRNG based on HMAC

- **TLS/WTLS**

```
m = $\lceil n/outlen \rceil$
A(0) = V; W = null;
for i = 1 to m
    A(i) = $HMAC_k(A(i-1))$;
    $w_i = HMAC_k(A(i) || V)$;
    W =|| $w_i$;
return msb(W, n)
```