Server
(n,d)

```
send-flag (flag, signature)
get-flag-ids : ids of last x minutes
check-flag (id)
    receive enc-Key, enc-flag
    Key = $enc\text{-}Key^1 \% n$
    $K_1$ = Key[-56:]
    $K_2$ = Key[-112:-56]
```

Wie erfährt Bambi das?

Bambi erhält Key ebenso per RSA verschlüsselt

Client
(n,e) $e = 17$, $n \sim 2048$ bit] Server Key

112 bit

DB : (flag-id, time, Key, enc(flag, Key)) ← maybe SQL injection to get Key

ensure no manipulation

enc - TripleDES K_1, K_2, K_1

```
send-flag :
  encode Key with RSA
  send ($Key^e$ mod n, enc-flag)
```

fix: pad Key with random bits only last are taken

attack: Key ~112 bit

$\Rightarrow Key^{17} < n \Rightarrow Key^e \% n = Key^e$

take root in $\mathbb{Z}$

```
init:
  generate RSA-Key
  using random-seed(const)
  → all have same Key
```

↖ own Key might get $e \neq 17$?

```
send pubKey
  send Key after query to server/player
  if none exists, generate via init
```