Cryptanalysis of EAX-Prime

Kazuhiko Minematsu, NEC Corporation
Stefan Lucks, Bauhaus-Universität Weimar
Hiraku Morita, Nagoya University
Tetsu Iwata, Nagoya University

DIAC, Directions in Authenticated Ciphers
July 5--6, 2012, Stockholm, Sweden

EAX-Prime (EAX')

- Authenticated encryption based on AES
- Standard security function for the Smart Grid
  - ANSI C12.22-2008
- proposed by Moise, Beroset, Phinney, and Burns to NIST in 2011
- NIST announcement:

  "Future Parts: NIST is planning to develop two additional parts to the 800-38 series of Special Publications. One will specify schemes for format preserving encryption based on the FFX framework, and the other will specify the EAX' mode for authenticated encryption, in support of Smart Grid."

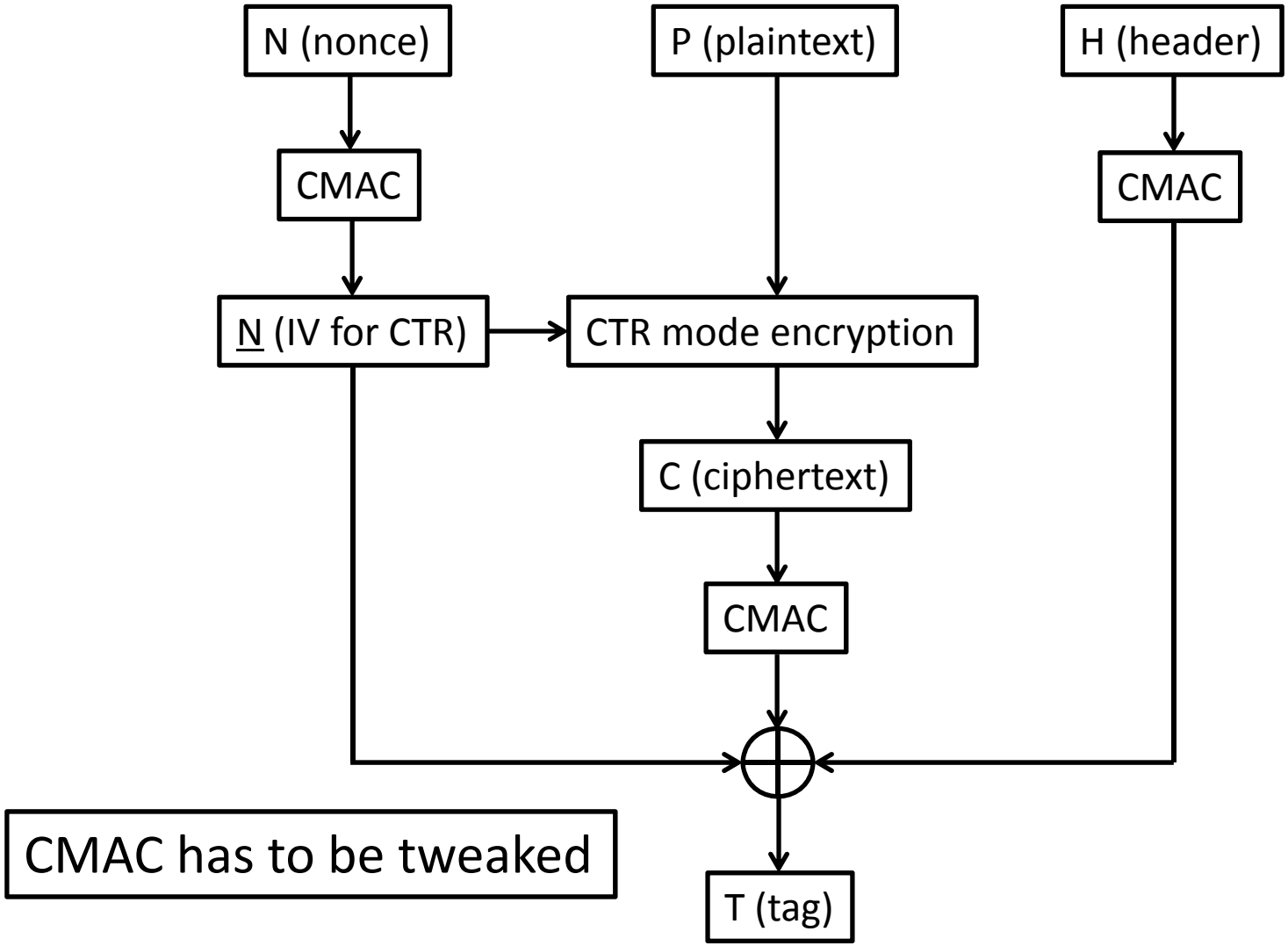
Overview of Our Results

- forgery attack
- chosen plaintext distinguisher
- chosen ciphertext message recovery attack

EAX and EAX-Prime

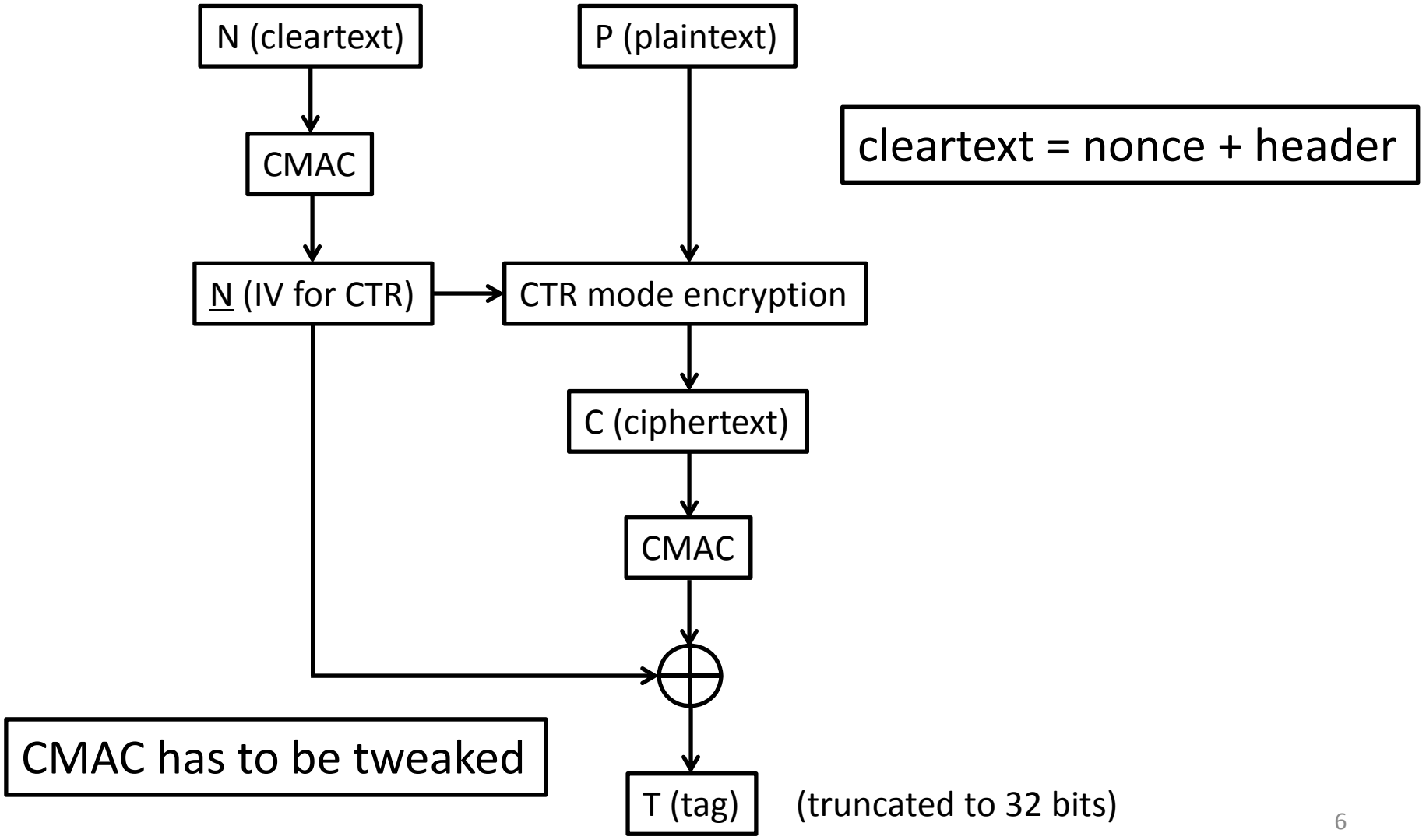
- EAX
  - an authenticated encryption proposed by Bellare, Rogaway, and Wagner at FSE 2004
  - has a proof of security
- EAX-prime
  - modified version of EAX to optimize the number of blockcipher calls and the size of memory
  - no formal analysis

EAX

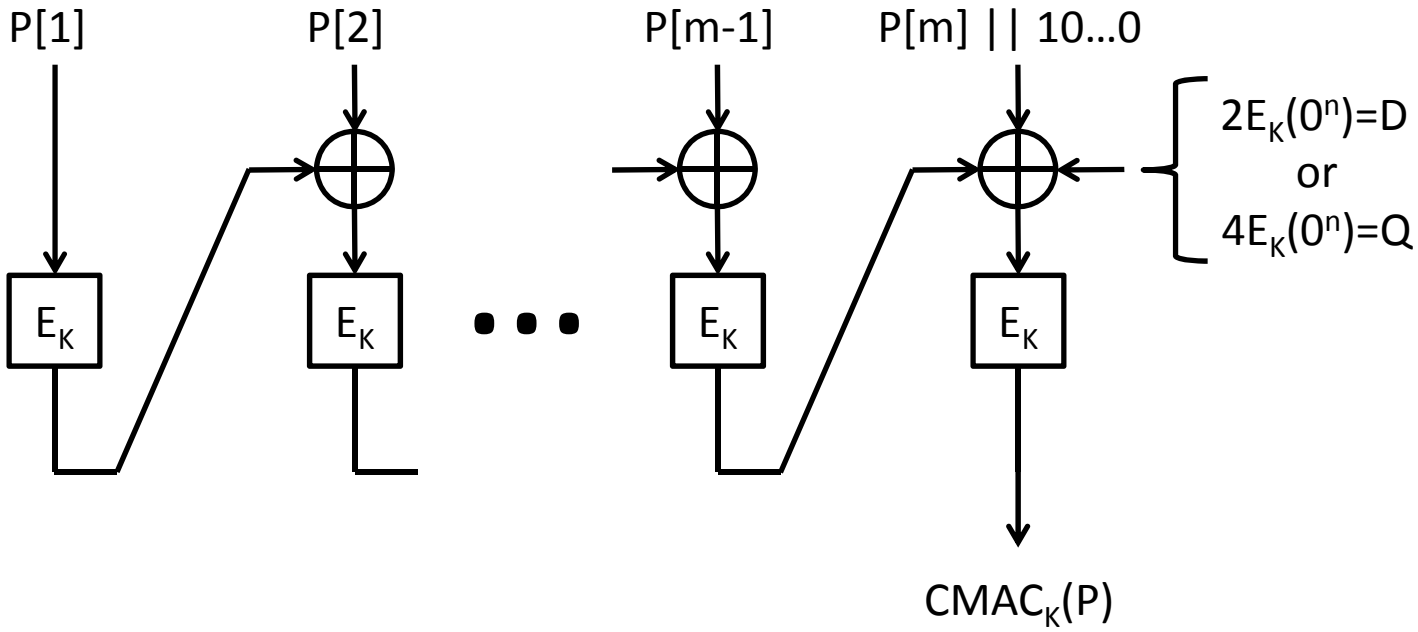


CMAC has to be tweaked

EAX-Prime

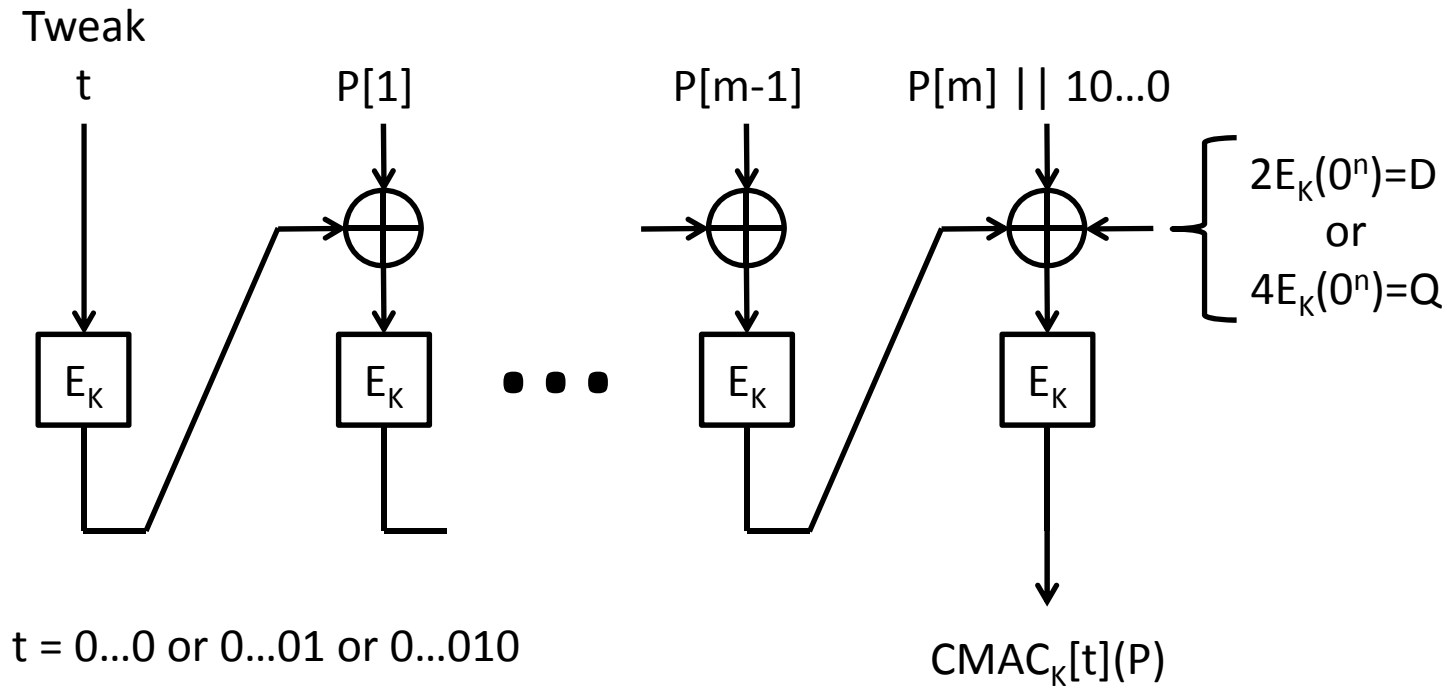

N (cleartext)
P (plaintext)
cleartext = nonce + header
CMAC
N (IV for CTR)
CTR mode encryption
C (ciphertext)
CMAC
CMAC has to be tweaked
T (tag)
(truncated to 32 bits)

CMAC [NIST SP 800-38B]

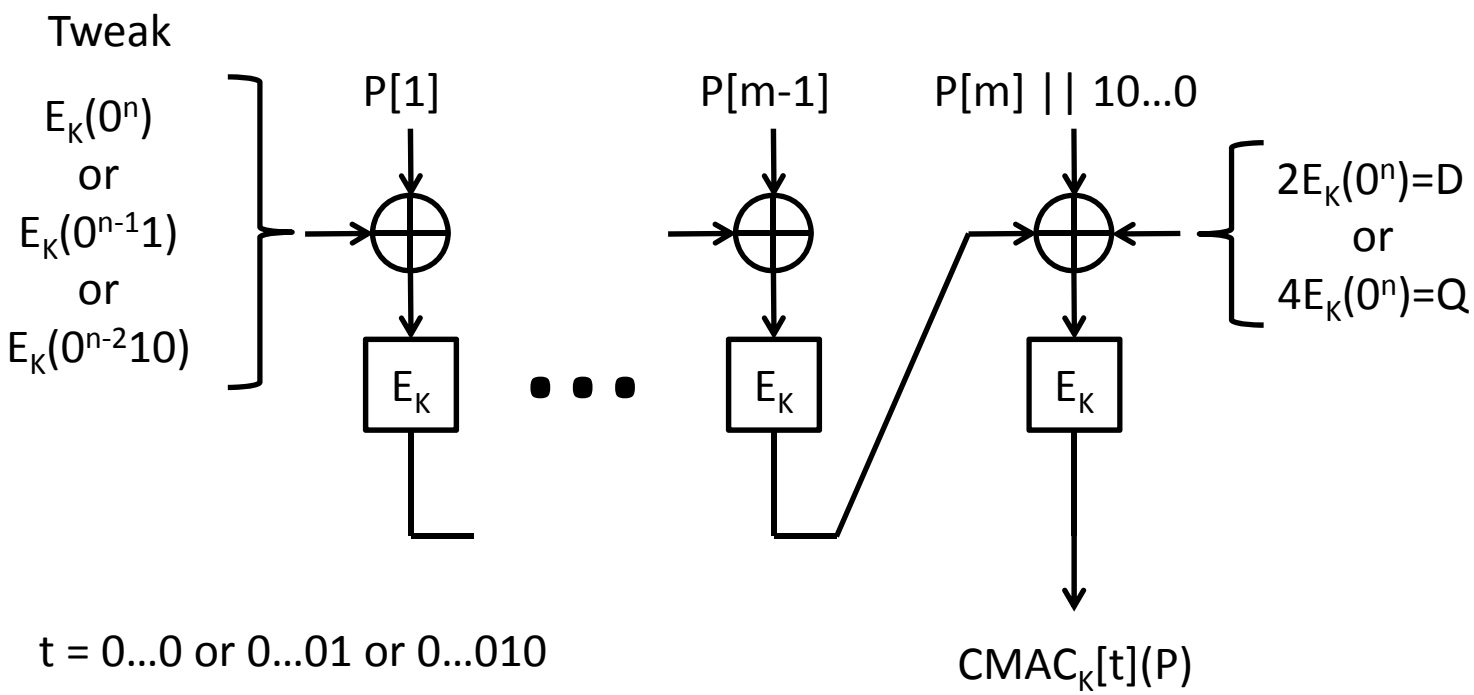


- MAC, variable-input length PRF
- $2E_K(0^n)$: ``doubling" of $E_K(0^n)$ in $GF(2^n)$
- $4E_K(0^n)$: $2(2E_K(0^n))$

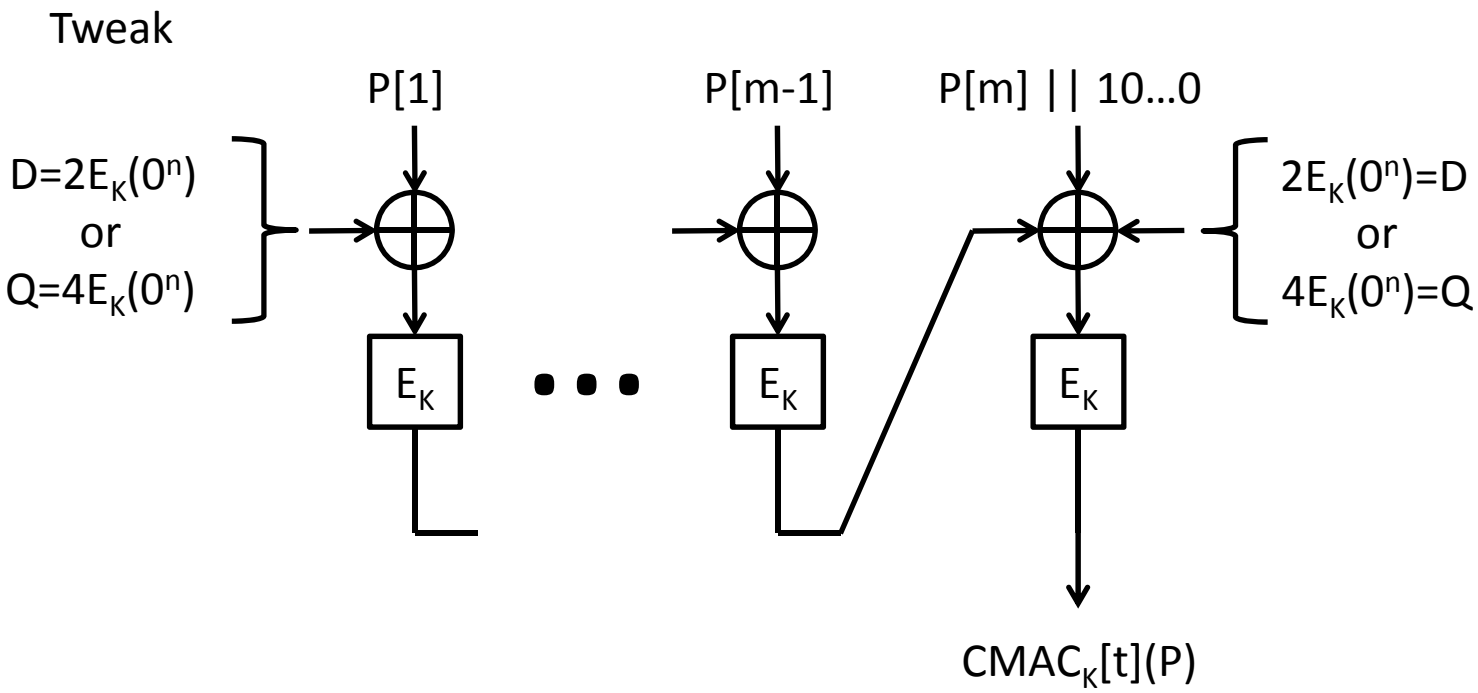
Tweaked CMAC in EAX



Tweaked CMAC in EAX

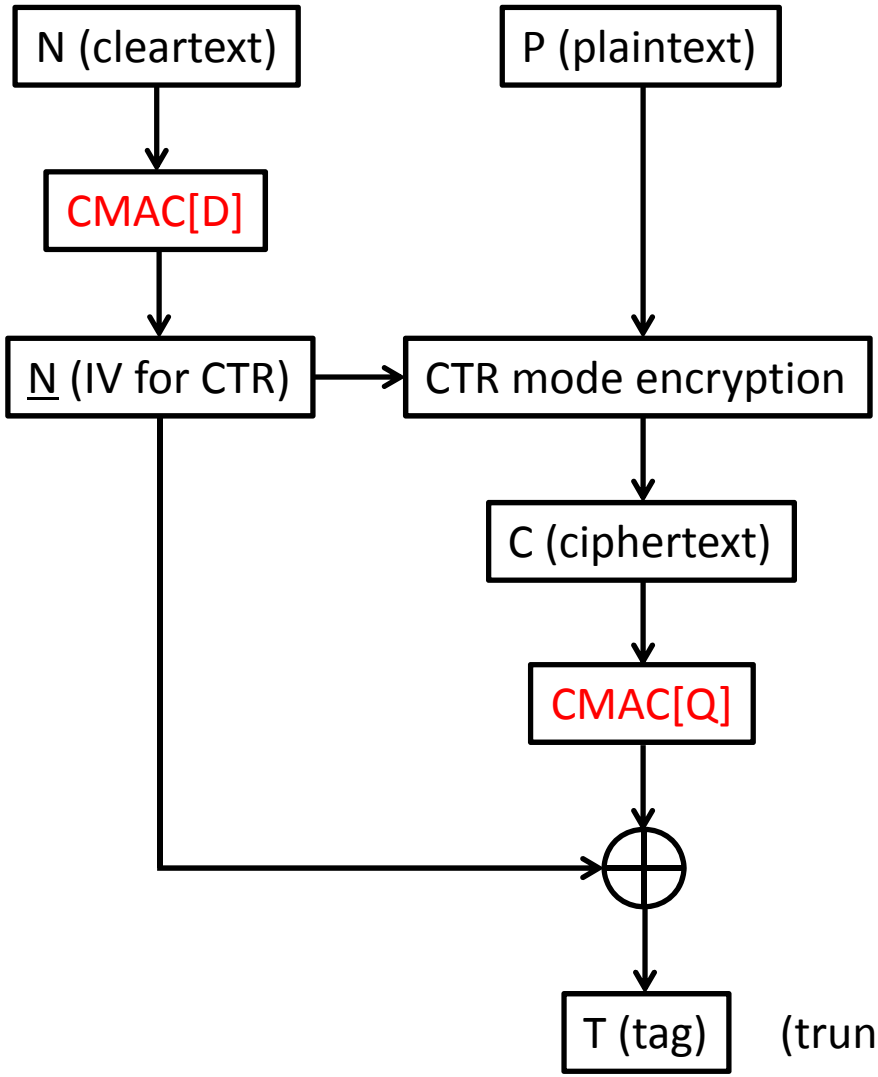


Tweaked CMAC in EAX-Prime



- CMAC[D]: Tweaked CMAC with $D=2E_K(0^n)$
- CMAC[Q]: Tweaked CMAC with $Q=4E_K(0^n)$

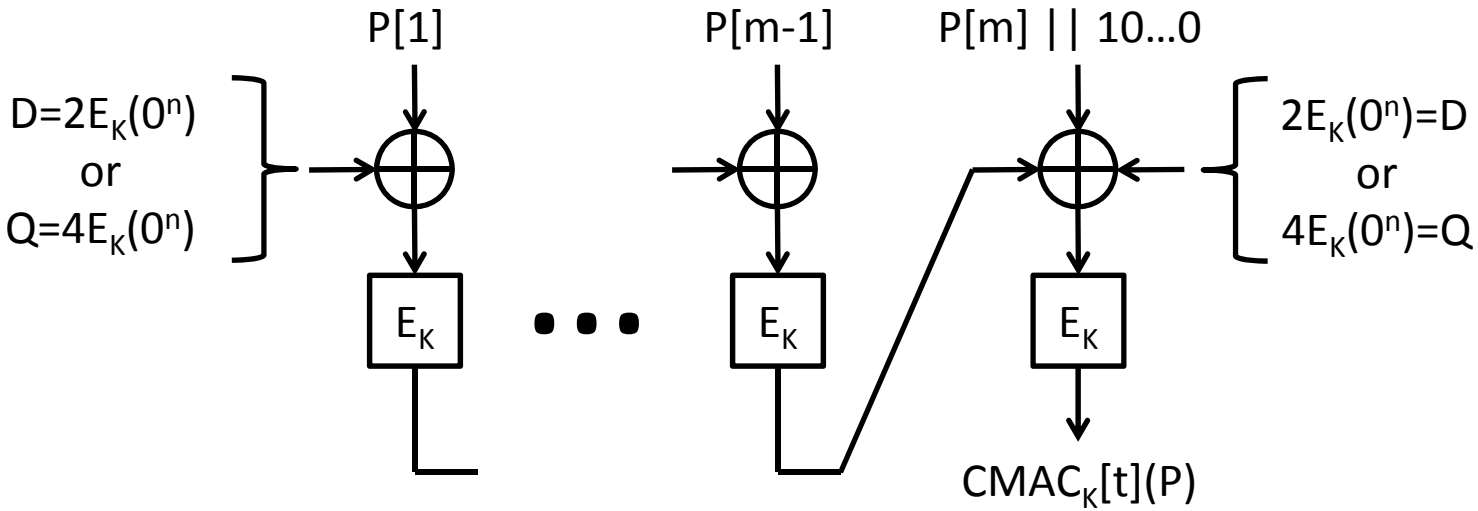
EAX-Prime



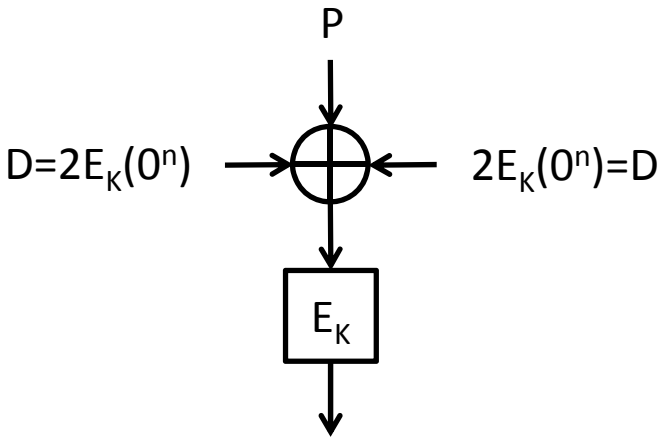
there are other minor changes

(truncated to 32 bits)

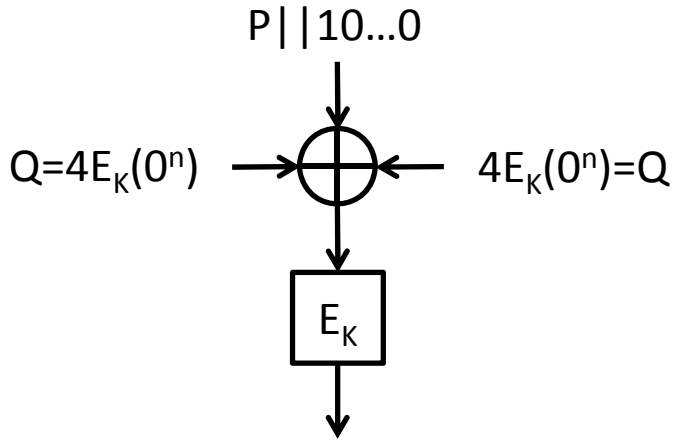
Observations on CMAC[D] and CMAC[Q]



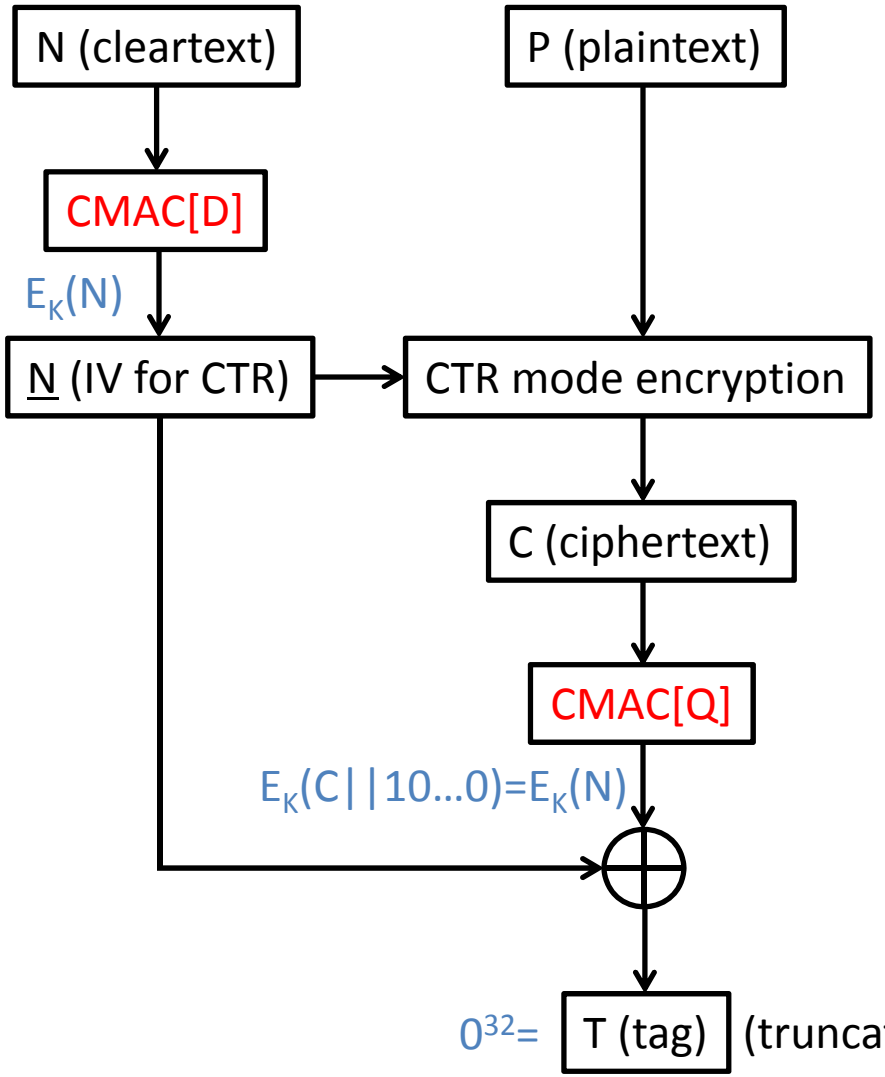
CMAC[D] when |P|=n



CMAC[Q] when 0≤|P|<n



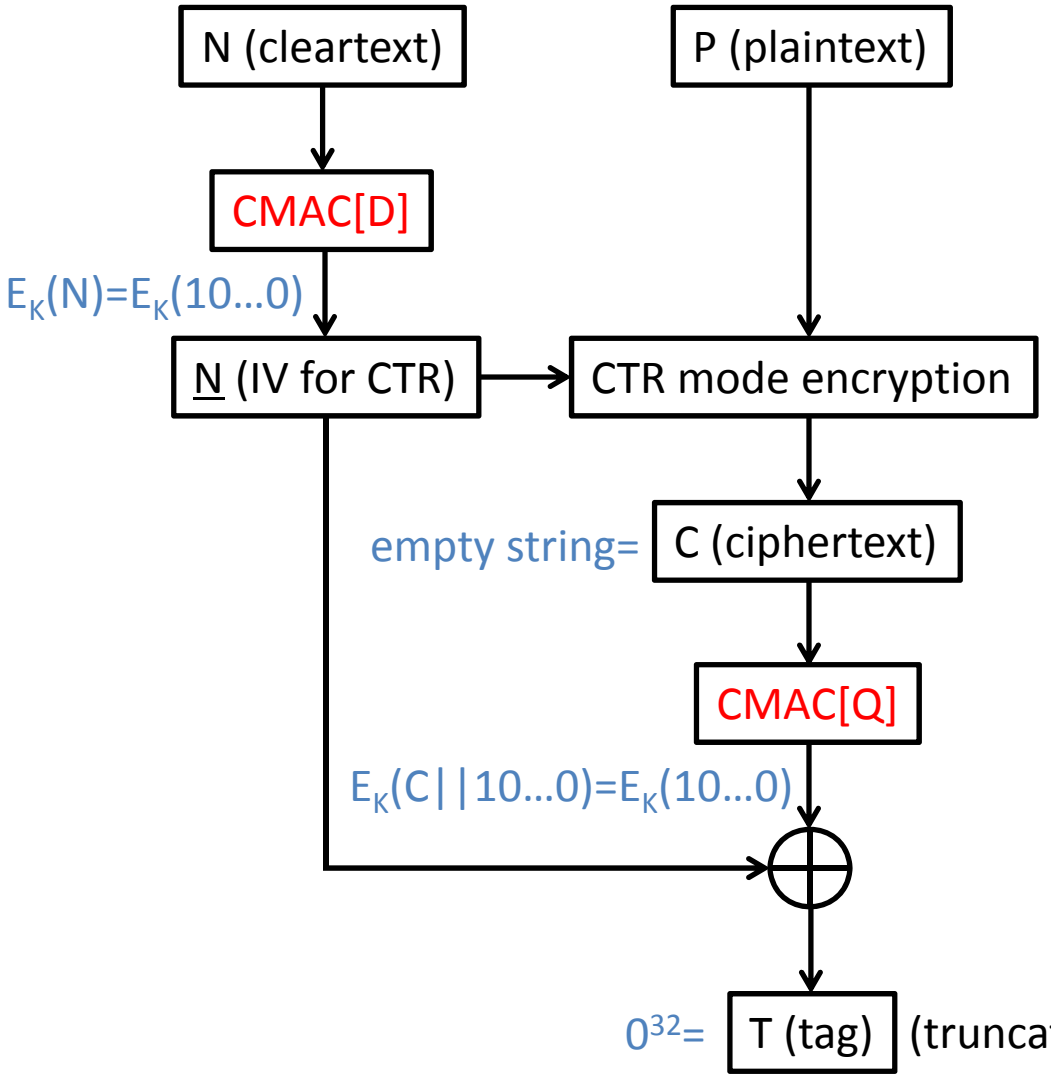
Forgery Attack



1. Let (N,C,T) be
   - $|N|=n$
   - $|C|<n$ and $C||10\ldots0 = N$
   - $T=0^{32}$
2. Ask (N,C,T) to the decryption oracle

succeeds with probability 1 (without making any encryption queries)

Distinguishing Attack

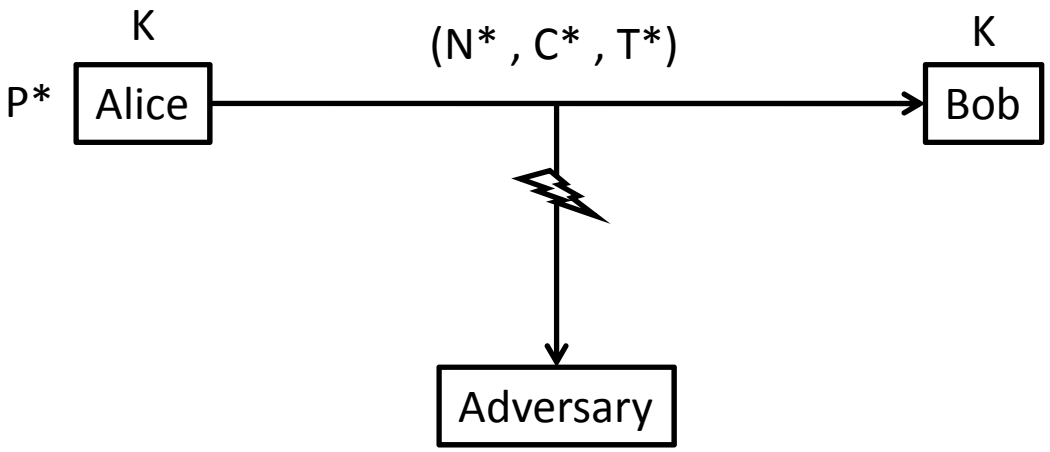


1. Let (N,P) be
   - N=10…0, |N|=n
   - |P|=0 (empty string)
2. Ask (N,P) to the encryption oracle and obtain (C,T)
3. output 1 if $T=0^{32}$
   output 0 otherwise

$T=0^{32}$ with probability
- 1 for the encryption oracle
- $1/2^{32}$ for the random oracle

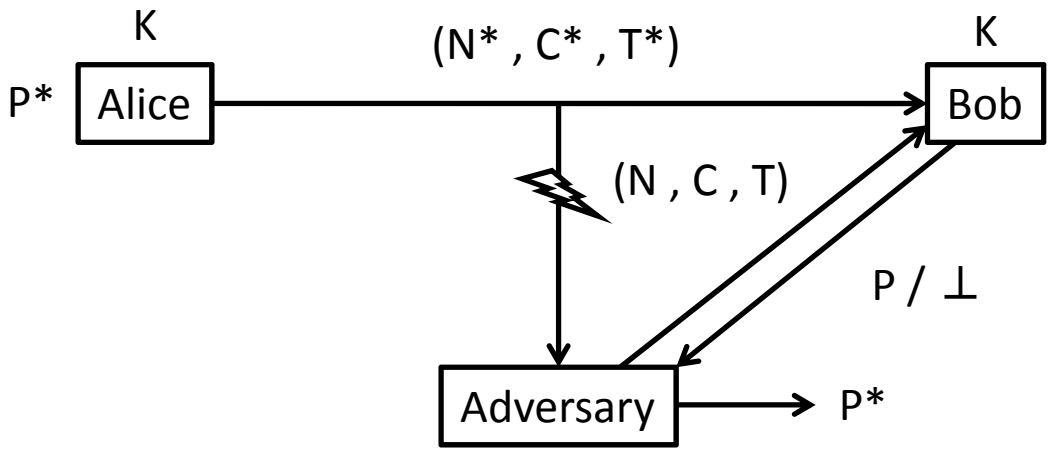
succeeds with a high probability (with one encryption query)

Chosen Ciphertext Message Recovery



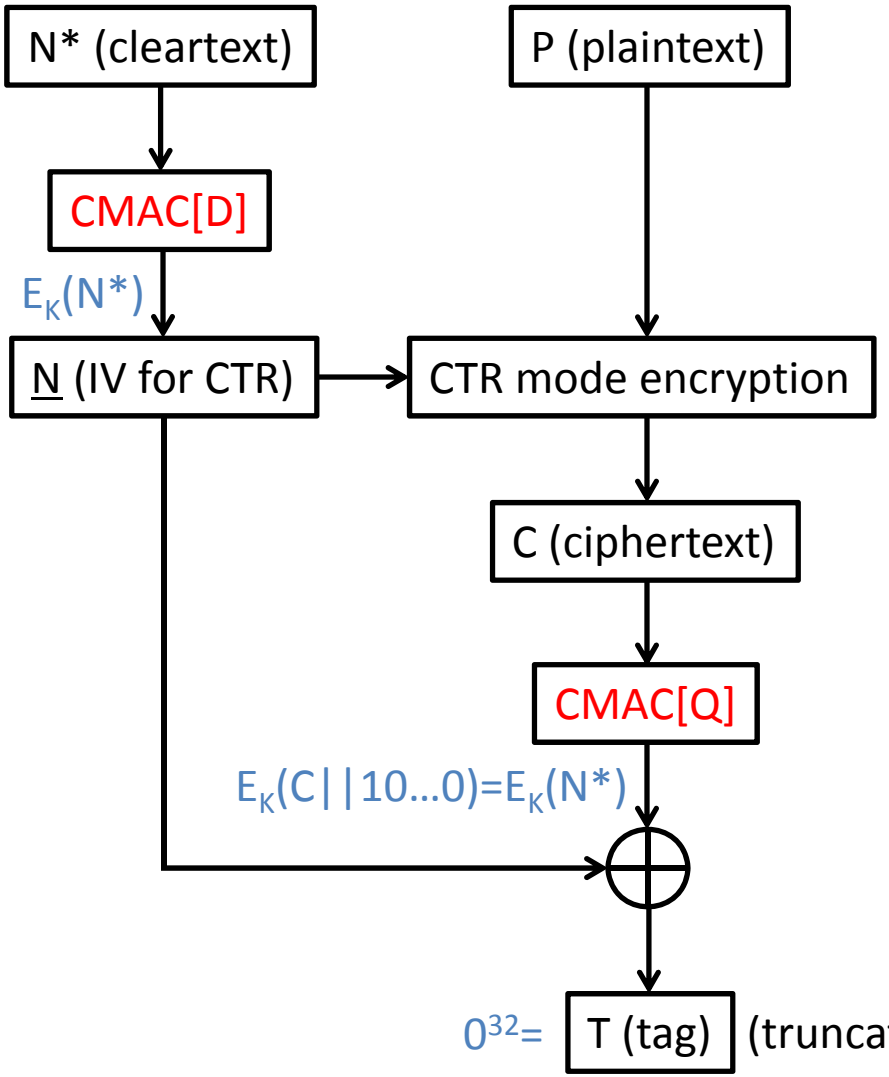
- Consider (N*, C* , T*)
  - the corresponding P* is unknown to the adversary
  - the adversary eavesdrops (N*, C* , T*)
- The adversary can ask (N , C , T) to a decryption oracle
- The goal is to find (a part of) P*

Chosen Ciphertext Message Recovery



- Consider (N*, C* , T*)
  - the corresponding P* is unknown to the adversary
  - the adversary eavesdrops (N*, C* , T*)
- The adversary can ask (N , C , T) to a decryption oracle
- The goal is to find (a part of) P*

Chosen Ciphertext Message Recovery



1. Suppose (N*,C*,T*) satisfies |N*|=n
2. Let (N, C, T) be
   - N=N*
   - |C|<n and C||10…0 = N*
   - $T=0^{32}$
3. Ask (N*, C , T) to the dec. oracle
4. Let P be the answer
5. C xor P is the keystream for N*

first |C| bits of P*
= (first |C| bits of C*) xor (C xor P)

succeeds with probability 1

Applicability to the ANSI C12.22 Protocol

- The attacks can be slightly generalized to handle other input lengths
- None of our attacks works if $|N| > n$
  - we do not know if $|N| > n$ is guaranteed in ANSI C12.22 specification
- The attacks can be avoided if $|N| > n$ is "guaranteed"
  - should be actively checked by the decryption side
  - even if $|N| > n$ is stated in the specification, this does not prevent a malicious adversary from using $|N| \leq n$

Practical Implication*

- EAX-prime is intended for smart grid applications
  - it hardly seems reasonable to assume that every device will always carefully check the lengths of the input data
- Forgery attacks allow a malicious adversary to create a large number of valid short messages
  - possibly result in random-looking commands
  - practical implication depends on what the actual device will do with valid and random commands

* Thanks to Greg Rose for discussions on this point.

Discussions

- What went wrong?
  - Compared to EAX (among other changes), EAX-prime changes the “key dependent constant”
    - reduces the number of blockcipher calls
  - This is generally a dangerous sign as the original scheme is usually designed to optimize the number of calls
  - Sometimes changing the “key-independent constant” may break the provable security result
    - e.g., in GCM, when |N| = 96, IV = N || 0…01
    - changing this to IV = N || 0…0 results in an insecure scheme
    - seemingly a minor modification may result in an insecure scheme

Discussions

- It seems difficult to formalize “what can safely be changed”
- General advice: If the existing scheme is modified,
  - entire security proof should be revisited (ask cryptographers)
  - or, do not modify the existing scheme

Conclusion and Open Question

- EAX-prime allows forgery attacks, chosen plaintext distinguishing attacks, and chosen ciphertext message recovery attacks
- The changes break the provable security results of EAX
  - EAX-prime is cryptographically broken as a general purpose authenticated encryption
  - Our attacks do not work on EAX (a proof of security)
- Open question:
  - prove or disprove the security of EAX-prime if $|N| > n$