Midgame Attacks
(and their consequences)

Donghoon Chang[1] and Moti Yung[2]
[1]IIIT-Delhi, India
[2]Google Inc. & Columbia U., USA

Crypto is a Technical Science

- As technology moves, so should crypto designs and constructions!
- As technology moves → new attack vectors become feasible or practical.
- We have to constantly be aware of and also be prepared for changes in technologies and in related attack scenarios.

- → This is one reason why cryptographers do not sleep well ☹ …but also why they get jobs ☺ !

Midgame Attacks

- **At some point in the middle of computation with a secret key (midgame), and after some secure work (typically initial work), the powerful adversary sees the entire internal state and attempts key recovery/ forgery/ decrypting.**

- **For cloud delegated work, hide long term key from provider (after performing small work):**

  E.g., HMAC when the first & second "key hashing" is applied @user… while the rest of the heavy (bulk) hashing work can be performed @cloud starting from an intermediate state.

Motivation

- **Cloud Computing→ Secure Delegation→** No need to give away keys to the cloud, just a midgame state (i.e., **local rather than global crypto-work delegation**).→ better privacy!

- There is **no perfect Security Guarantee** even in Cryptographic Modules (assume attack at some point in the computation, and assume full leakage at time of attack). [in other areas: forward secrecy, key insulated mitigation was considered but not in basic designs!]

Midgame vs. Side-channel Attacks

- Side-channel Attacks
  - Non-invasive & Passive Attacks
  - Power Analysis
- Midgame Attacks
  - Invasive Attacks
  - Memory Dump Attack (as cold boot attack but 100% disclosure;

    goal is to compartmentalize the damage)

Midgame vs. Leakage Attacks

- Leakage Attacks
  - Partial Information is leaked
  - Gives leakage-resilient Cryptographic Models
- Midgame Attacks
  - Total leakage at some point

Note that once a partial information is leaked, then usually it brings total leakage by the divide-and-conquer attack strategy on a symmetric key. So, the total leakage assumption at some point is practically-justified while partial leakage assumption has been criticized by some practitioners.

Summary

- **Concrete Midgame Attacks:**
  - **many known block-cipher encryption schemes and modes are not secure.**
  - **six ECRYPT stream ciphers, except Rabbit, are not secure.**
  - **HMAC-Keccak, unlike other four SHA-3 finalists, is not secure; first security gap among the 5.**

Overall: This is more about new issues/ notions/ revised design rules & not about technicalities of the relatively simple but demonstrative attacks.

Midgame Attacks on Block Cipher-based Encryption Schemes

- ECB, CBC, OFB, CFB, CTR Modes of Operation (approved by NIST) and many other encryption modes
- CCM, GCM (approved by NIST), OCB, and many other authenticated-encryption modes
  - **During the entire process of encryption, the secret key is fixed for every block cipher call**, so the key-recovery attack is possible in the midgame attacks.

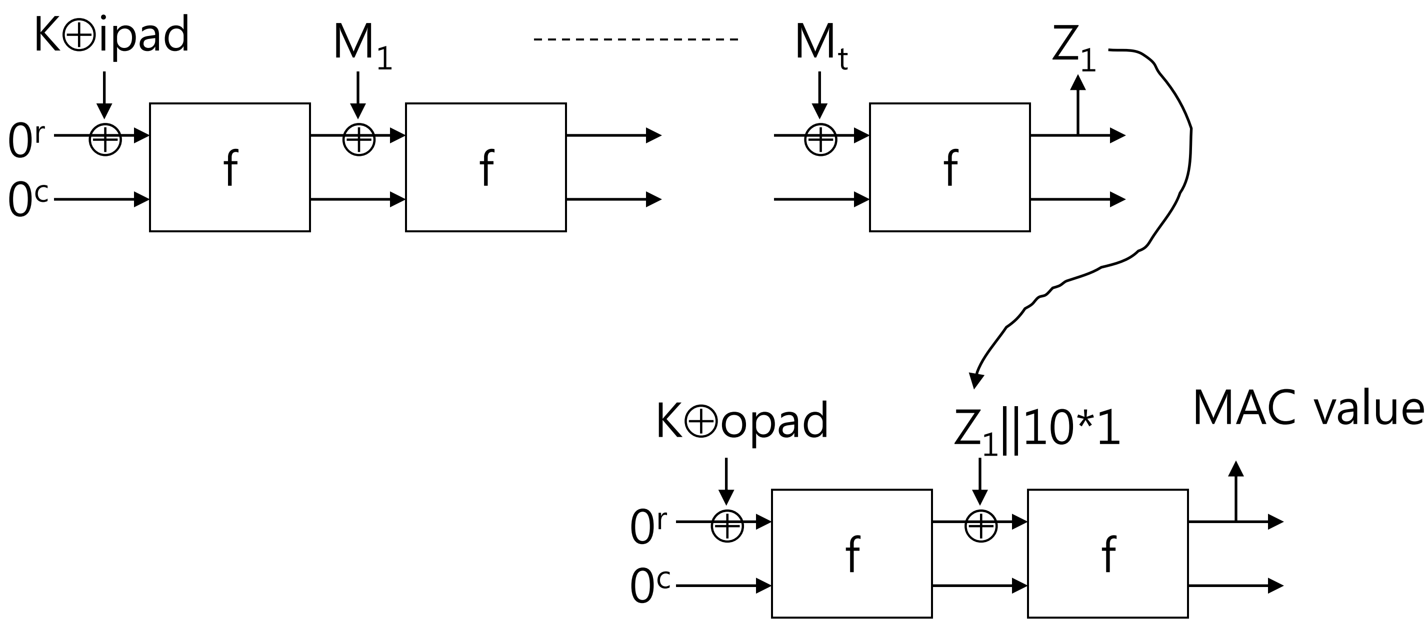
Midgame Attacks on Six ECRYPT Stream Ciphers

- There are Seven ECRYPT Stream ciphers.
  - Except Rabbit, all the other six stream ciphers are not secure against midgame attacks.
  - Except Rabbit, all **the internal computations are invertible.**
  - Once a midgame attacker knows any internal state, then he can generate all the previous key stream of the Six stream ciphers.

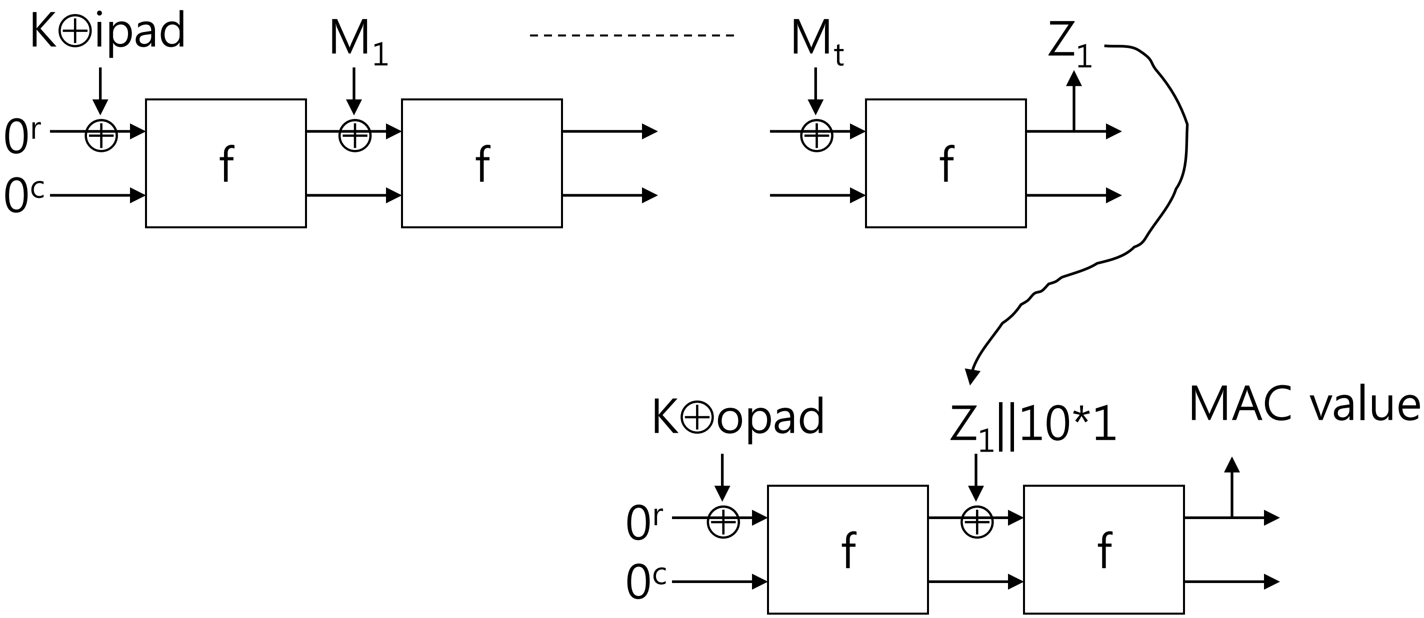
Midgame Attacks on HMAC based on the SHA-3 Finalists

- There are Five SHA-3 Finalists.
  - Except Keccak, all the other four SHA-3 final candidates provide better security against midgame attacks.
  - Keccak uses a simple domain extension, called **Sponge construction, which is based on an invertible permutation**, so it is easy to compute the key of HMAC once any internal state is leaked.

HMAC-Keccak (for one-block K)



HMAC-Keccak (for one-block K)



If we know *any internal state*, we can compute the key K because f is efficiently invertible.

Conclusions

- **Typical cryptographic schemes were designed without considering midgame security (since the notion is new !! Cloud-motivated).**
- **Designing new schemes, secure against midgame attacks [under new design rules] is a new direction (we have some designs). This includes formalizing security..**
- **Midgame analysis can be applied to numerous other areas such as public-key cryptography.**
- **Intuitively: For strong midgame security, locally-one-way & locally-pseudorandom operations should be considered which are fast for efficiency (just being fast is not enough).**

Therefore… remember:

"end of crypto" reported on Monday's invited talk……..

but!! Crypto is a Phoenix