Slide Attacks on a Class of Hash Functions

Michael Gorski[1] **Stefan Lucks**[1] **Thomas Peyrin**[2]

[1]Bauhaus-University of Weimar

[2]Orange Labs and University of Versailles

A n-bit block cipher E with r rounds is split into b identical rounds of the same keyed permutation F^i for $i = \{1, \ldots, b\}$:

$$\begin{aligned} E &= F^1 \circ F^2 \circ \cdots \circ F^b \\ &= F \circ F \circ \cdots \circ F \end{aligned}$$

A plaintext P_j is then encrypted as:

$$P_j \xrightarrow{F} X^{(1)} \xrightarrow{F} X^{(2)} \xrightarrow{F} \ldots \xrightarrow{F} X^{(b-1)} \xrightarrow{F} C_j.$$

Slide Attacks for Block Ciphers

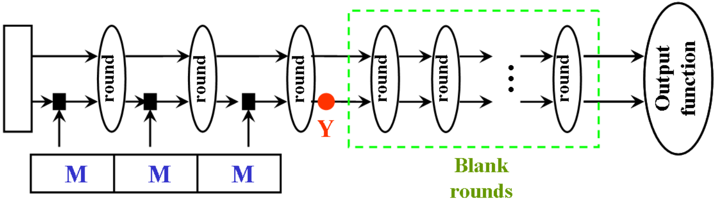
To mount a slide attack one has to find a slid pair of plaintexts (P_i, P_j), such that $P_j = F(P_i)$ and $C_j = F(C_i)$ holds.

$$P_i \xrightarrow{F} X^{(1)} \xrightarrow{F} X^{(2)} \xrightarrow{F} X^{(3)} \xrightarrow{F} \ldots \xrightarrow{F} C_i$$

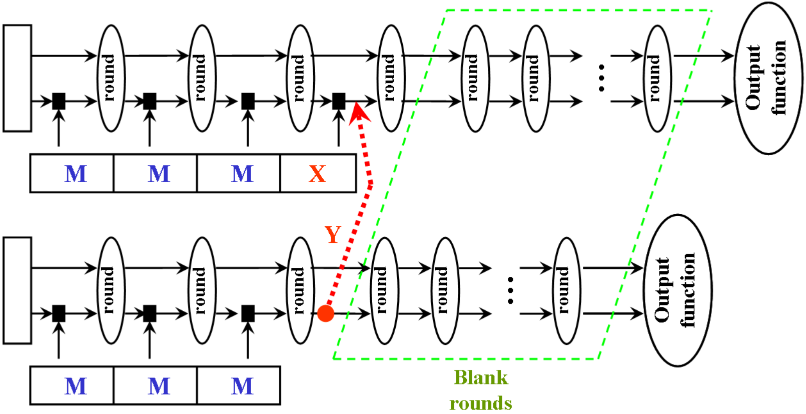
$$P_j \xrightarrow{F} X^{(2)} \xrightarrow{F} X^{(3)} \xrightarrow{F} \ldots \xrightarrow{F} X^{(b-1)} \xrightarrow{F} C_j$$

With the birthday paradox, only $2^{n/2}$ plaintexts are required to find a slid pair.

Application of slide attacks against hash functions were very few studied (Saarinen applied slide attacks against the inner cipher of SHA-1).


round
round
round
round
round
round
Output function
Y
M
M
M
Blank rounds

If the addition of X is neutral, then $output1 = round(output2)$.



What can we obtain from slide attacks ?

- slide attacks are a typical block cipher cryptanalysis technique.
- doesn't seem useful for collision or preimage attacks ...
- ... but we can "distinguish" the hash function from a random oracle.
- the key recovery attack may also be useful if some secret is used in the hash function: we can attack a MAC construction using a hash function.

We'll try to attack the following MAC construction:

$$\mathrm{MAC}(K, M) = H(K||M).$$

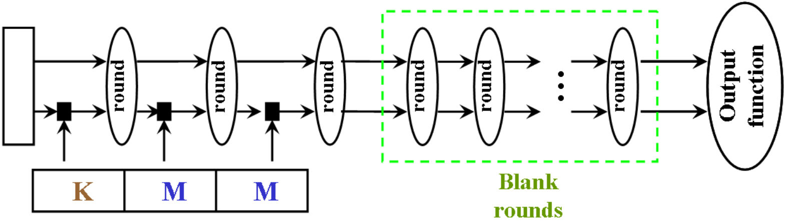
We'll try to attack the following MAC construction:

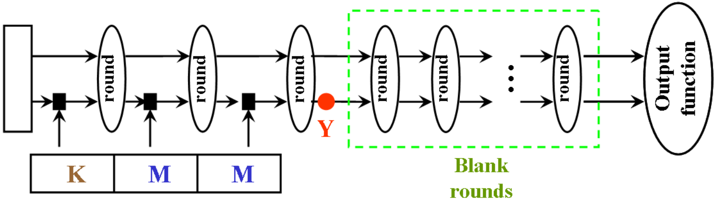
$$\mathrm{MAC}(K, M) = H(K||M).$$

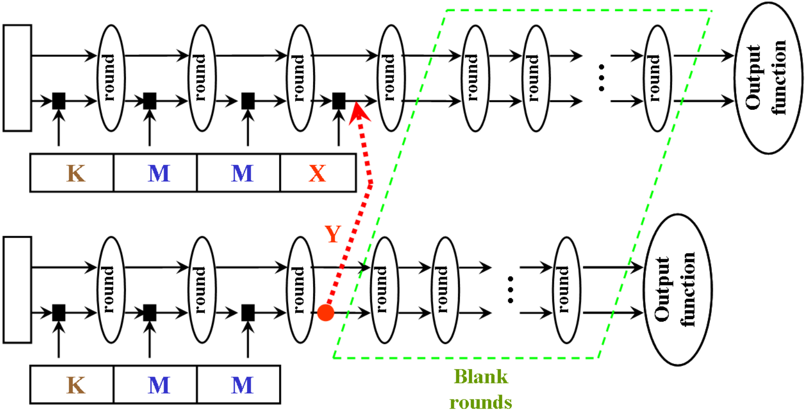
- **... which is secure if the hash function is modeled as a random oracle.**
- **Merkle-Damgård already known to be weak against this construction**: given $\mathrm{MAC}(K, M) = H(K||M)$, compute $\mathrm{MAC}(K, M||Y) = H(K||M||Y)$ without knowing the secret key K.
- patch provided in Coron *et al.*'s paper from Crypto 2005.

$$\text{MAC}(K, M) = H(K||M).$$

HMAC would be very slow with a sponge function, due to the blank rounds. Thus, the authors advised the following MAC construction:




round
round
round
round
round
round
Y
K
M
M
Blank
rounds
Output
function

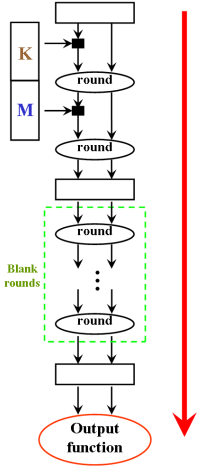

round
round
round
round
round
round
round
Output function
K
M
M
X
Y
round
round
round
round
round
round
Output function
K
M
M
Blank rounds

The Attack Scenario: the attacker makes queries M_i and receives $H(K||M_i)$. He then tries to get some non trivial information from the secret K or manage to forge another MAC with good probability.

The attack will be in three steps:

- Find and detect slid pairs of messages.
- Recover the internal state.
- Uncover some part of the secret key (or forge a new MAC).

The padding must also be taken in account !

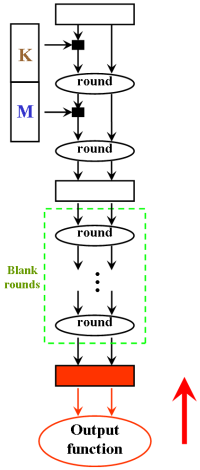


The Attack Scenario: the attacker makes queries M_i and receives $H(K||M_i)$. He then tries to get some non trivial information from the secret K or manage to forge another MAC with good probability.

The attack will be in three steps:

- Find and detect slid pairs of messages.
- Recover the internal state.
- Uncover some part of the secret key (or forge a new MAC).

The padding must also be taken in account !

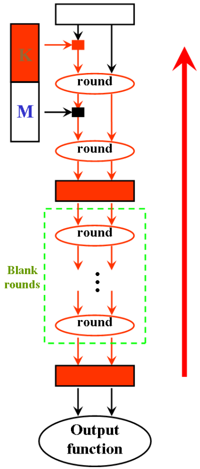


The Attack Scenario: the attacker makes queries M_i and receives $H(K||M_i)$. He then tries to get some non trivial information from the secret K or manage to forge another MAC with good probability.

The attack will be in three steps:

- Find and detect slid pairs of messages.
- Recover the internal state.
- Uncover some part of the secret key (or forge a new MAC).

The padding must also be taken in account !



Find a slid pair of messages:

- depends on the message insertion function.
- impossible in the original sponge framework (in which the last inserted word must be different from 0) ...
- ... but possible if a different padding is used !
- possible if the insertion function overwrites the corresponding internal state words (as in GRINDAHL) with $P = 2^{-r}$.

Detect a slid pair of messages:

- depends on the output function.
- very easy with the sponge squeezing process (all the output words are shifted by one iteration position).
- more complicated with a direct truncation after the blank rounds.

Recovering the internal state and **uncovering the secret key** both depend on the whole hash function (require a case by case analysis).

Why not attacking

- HMAC ?
- or $\text{MAC}(K, M) = H(M||K)$?
- or $\text{MAC}(K, M) = H(K||M||K)$?

Because we need direct access to the last inserted word in order to get a slid pair.

It is very easy (and costless) for the designers to protect themselves against slide attacks.

If you're inserting message blocks with a XOR:

- just use exactly the sponge framework and make sure that the last inserted message word is different from zero.

If you're inserting message blocks by overwriting the corresponding internal state words:

- add a constant to the internal state just before the blank rounds to clearly separate them from the normal rounds.
- use a different transformation during the blank rounds.

For GRINDAHL**-256**, the attack allows to:

- distinguish from RO with 2^{64} queries and computation time.
- forge valid MACs or to recover 1 new byte of the secret with 2^{64} queries and 2^{80} computations.

For GRINDAHL**-512:** the attack allows to (**first cryptanalytic results on this version**):

- distinguish from RO with 2^{64} queries and computation time.
- forge valid MACs or to recover 4 new bytes of the secret with 2^{64} queries and 2^{80} computations.

For RADIOGATÚN**: attack doesn't apply**, but would work on an overwrite version of it.