A Generic Approach to Invariant Subspace Attacks

Cryptanalysis of Robin, iSCREAM and Zorro

Gregor Leander[1], Brice Minaud[2], Sondre Rønjom[3]

[1] Ruhr-Universität Bochum, Germany
[2] ANSSI and Université Rennes 1, France
[3] Nasjonal Sikkerhetsmyndighet, Norway

EUROCRYPT 2015

Plan

1. Introduction: invariant subspace attacks.
2. Finding invariant subspaces: a generic algorithm.
3. Results on Robin, iSCREAM and Zorro.
4. Commuting linear maps in Robin and Zorro.
5. Conclusion.

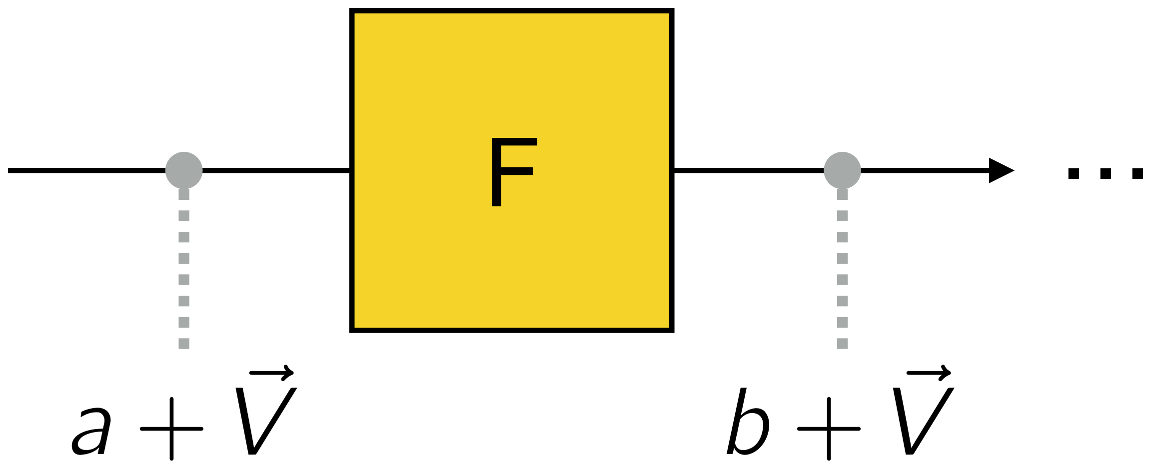
Invariant Subspace Attacks

Invariant Subspace Attacks were introduced at CRYPTO 2011.

Used to break PRINTCIPHER in practical time [LAKZ11].

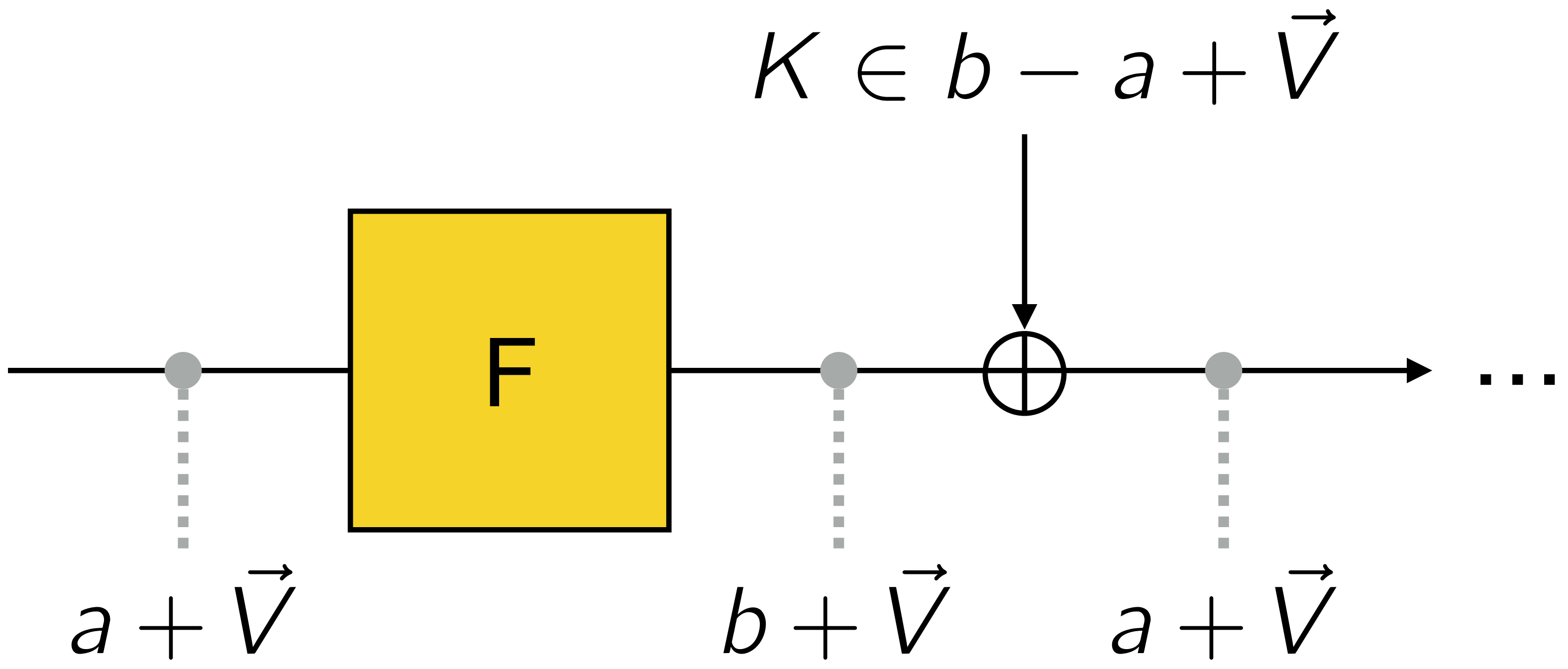
Take advantage of weak key schedules.

Invariant Subspace Attacks



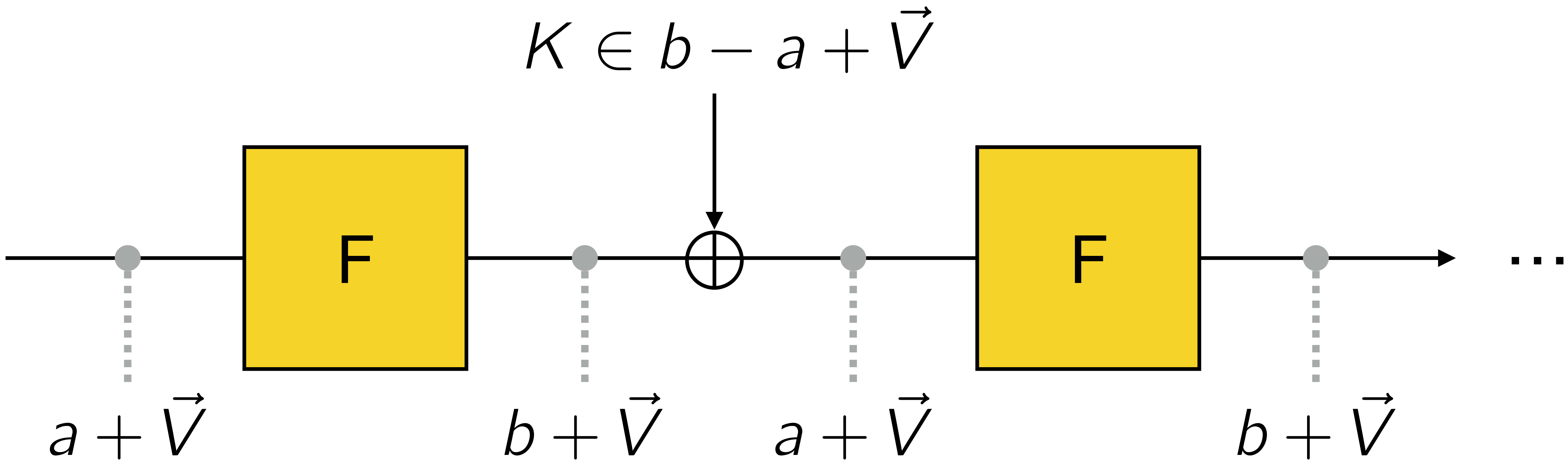
Assume the round function sends a some affine space to a coset of the same space.

Invariant Subspace Attacks



Now assume $K \in b - a + \vec{V}$...

Invariant Subspace Attacks

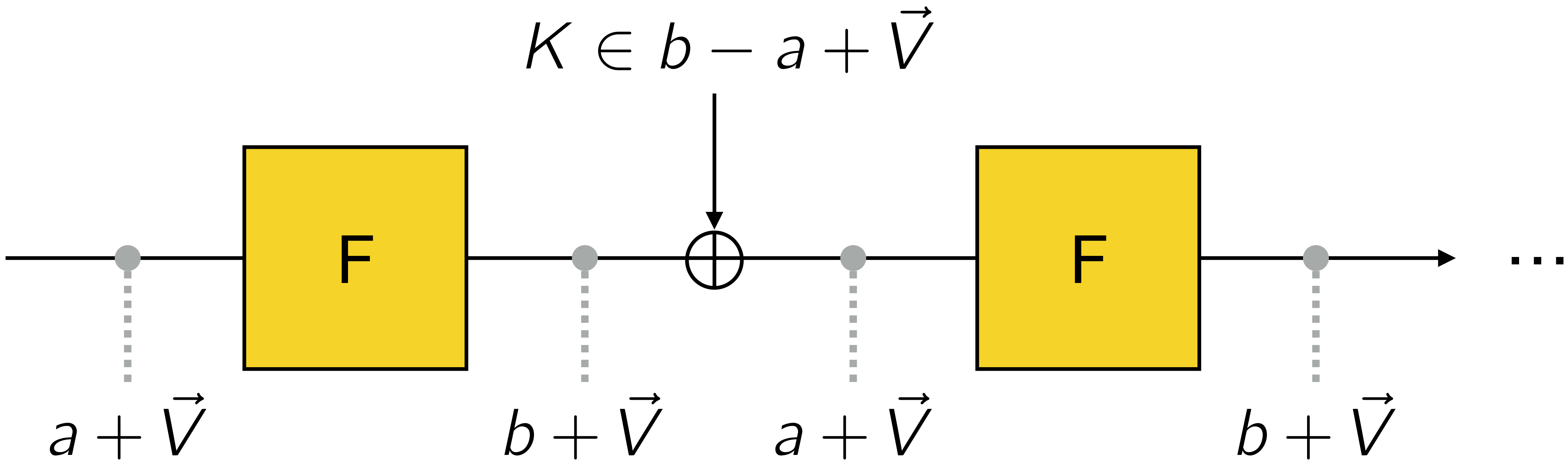


Now assume $K \in b - a + \vec{V}$...

Then this process repeats itself.

Plaintexts in $a + \vec{V}$ are mapped to ciphertexts in $b + \vec{V}$

Invariant Subspace Attacks

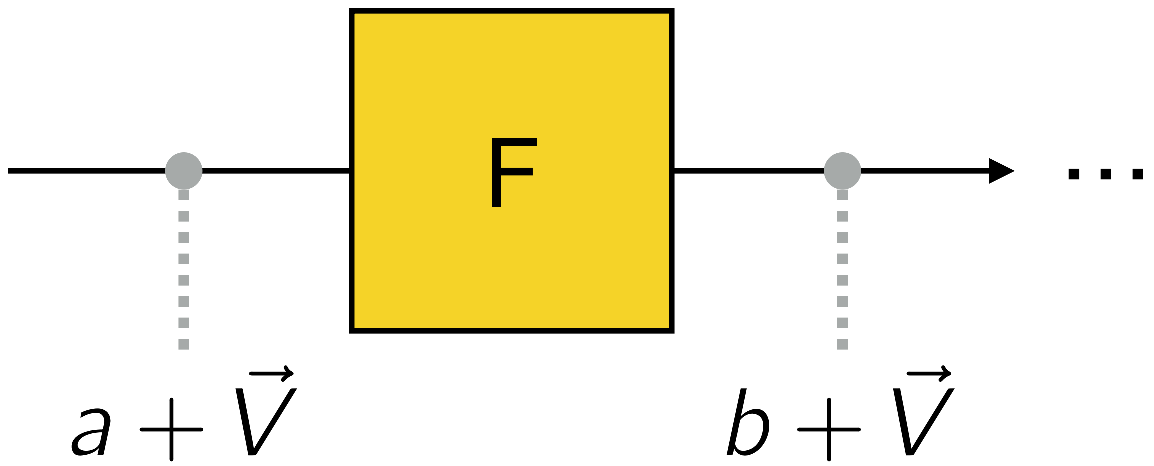


Confidentiality is broken.

Density of weak keys: $2^{-\text{codim}\,\vec{V}}$

Finding invariant subspace attacks: a generic algorithm

A Generic Algorithm

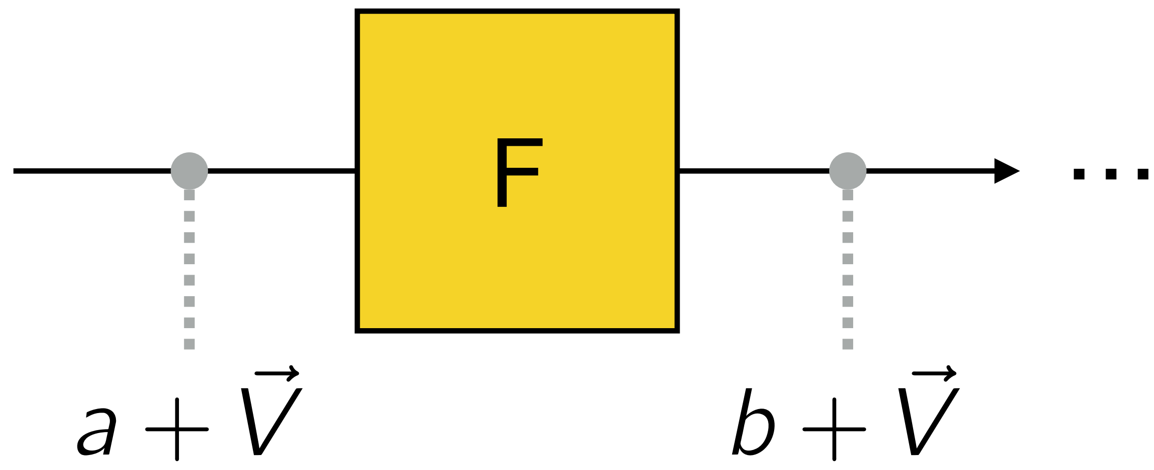


Bootstrap: assume we know $s, t \in a + \vec{V}$

Then $F(s), F(t) \in b + \vec{V}$ so $F(s) - F(t) \in \vec{V}$

Now we know one more vector of $\vec{V}$.

A Generic Algorithm



<u>**“Closure”** Algorithm</u>

Input: $s, \vec{W}$ such that $s + \vec{W} \subseteq a + \vec{V}$

Output: $a + \vec{V}$

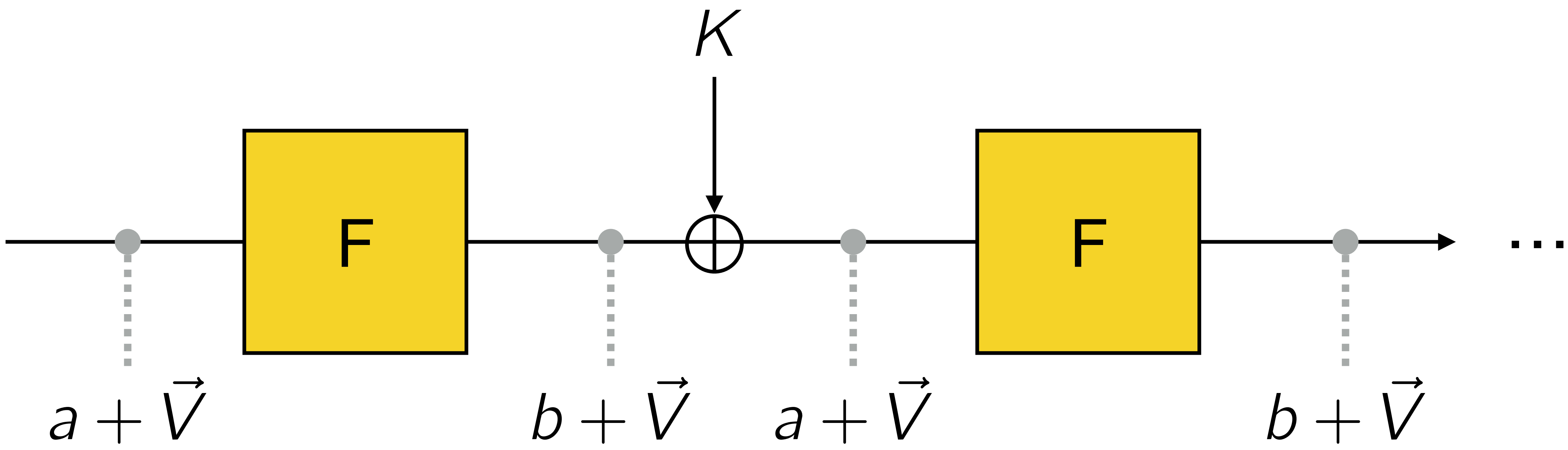
1. Pick $w \leftarrow_{\$} \vec{W}$
2. Add $F(s+w) - F(s)$ to $\vec{W}$
3. Iterate steps 1 and 2 until $\vec{W}$ remains stable for N iterations.
4. Return $s + \vec{W}$

A Generic Algorithm

A few remarks…

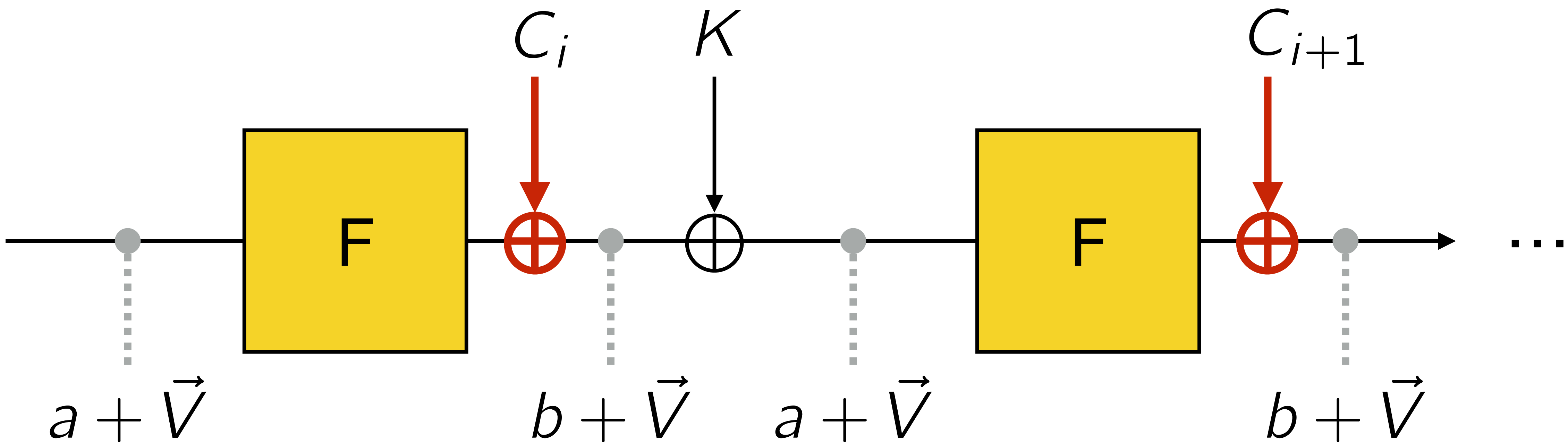
- The algorithm only outputs the smallest invariant subspace containing the input.
- … we still need to bootstrap.

Bootstrapping the Algorithm



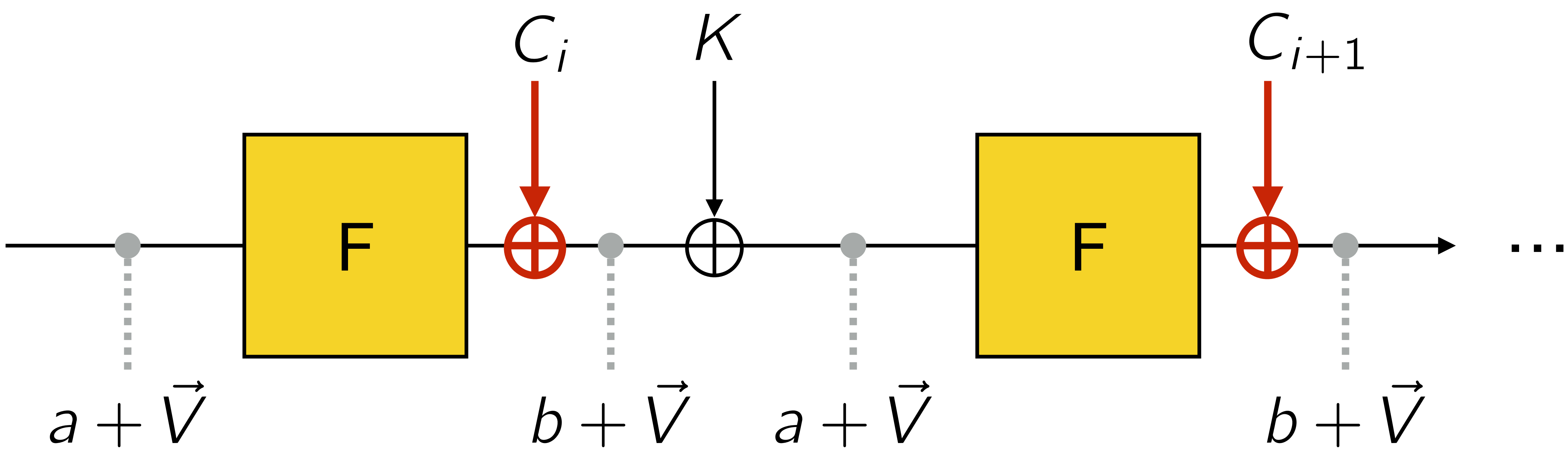
We cheated a little.

Bootstrapping the Algorithm



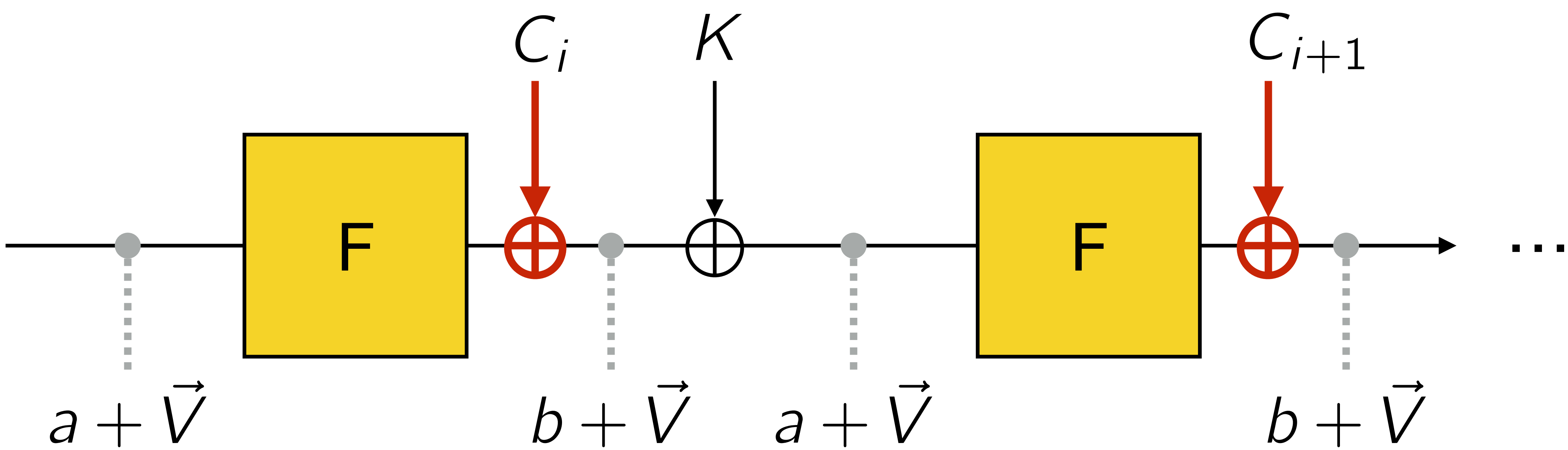
We cheated a little.

Bootstrapping the Algorithm



We really want $\forall i, C_i \in \vec{V}$

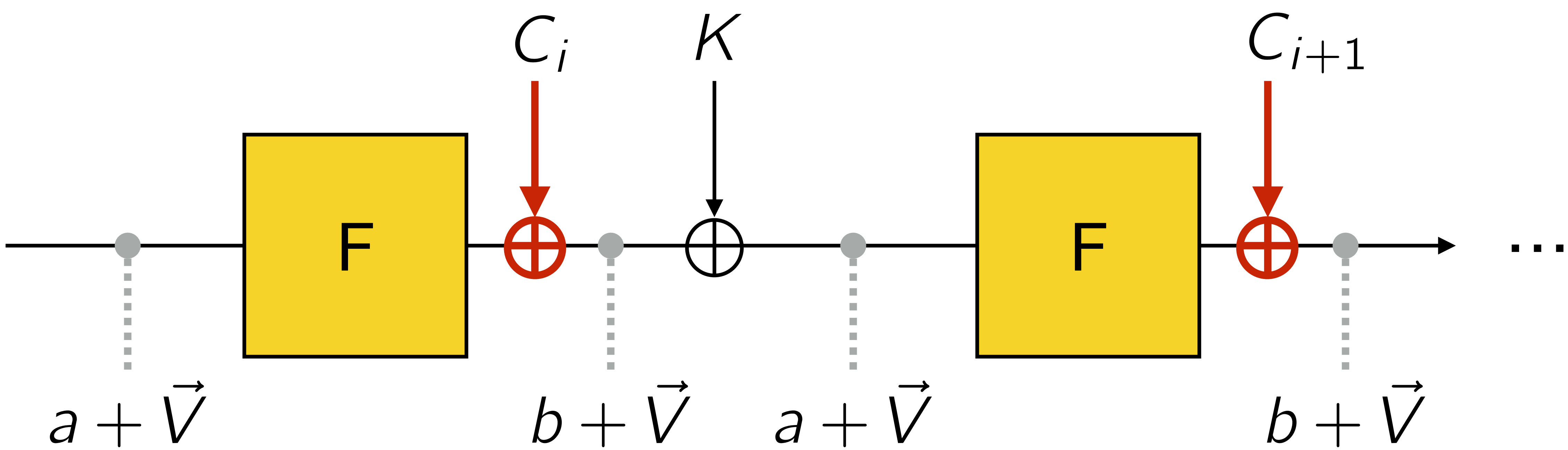
Bootstrapping the Algorithm



We really want $\forall i, C_i \in \vec{V}$

This gives us a "nucleon" $\vec{W} = \mathrm{span}\{C_i\} \subseteq \vec{V}$

Bootstrapping the Algorithm



We really want $\forall i, C_i \in \vec{V}$

This gives us a “nucleon” $\vec{W} = \mathrm{span}\{C_i\} \subseteq \vec{V}$

If $a \neq 0$, it remains to find an offset $s \in a + \vec{V}$.
We simply try many random offsets.

Complexity

<u>Generic Invariant Subspace Algorithm</u>

1. $\vec{W} \leftarrow \mathsf{span}\left\{C_i\right\}$
2. Guess offset s
3. Compute $\mathsf{Closure}(s + \vec{W})$
4. Repeat until $\mathsf{dim}(\mathsf{Closure}) < n$

Complexity

<u>Generic Invariant Subspace Algorithm</u>

1. $\vec{W} \leftarrow \mathsf{span}\,\{C_i\}$
2. Guess offset s
3. Compute $\mathsf{Closure}(s + \vec{W})$
4. Repeat until $\dim(\mathsf{Closure}) < n$

If $a + \vec{V}$ is actually a linear space : instant result.

Otherwise, on average: $2^{-\mathrm{codim}\,\vec{V}}$ tries.

Properties of the algorithm

- Generic: black-box use of round functions
- Does not disprove the existence of “small” spaces
- Public implementation: `http://invariant-space.gforge.inria.fr`

Results on Robin, iSCREAM and Zorro

Robin, iSCREAM and Zorro

Robin and Fantomas: lightweight ciphers, created to illustrate LS-designs, FSE 2014 [GLSV14].

SCREAM and **iSCREAM**: authenticated variants of Fantomas and Robin, CAESAR competition entries.

Zorro: lightweight cipher with partial nonlinear layer [GGNS13]. Broken by differential and linear attacks. Best attack: 2^{40} data/complexity [BDDLKT14].

Results on various ciphers

	Result	Running Time
Robin	**Subspace found!** codimension 32	22h
iSCREAM	**Subspace found!** codimension 32	22h
Zorro	**Subspace found!** codimension 32	<1h
Fantomas	With probability 99.9%: No invariant subspace of codimension < 32	
NOEKEON		
LED		
Keccak		

➡ Weak key set of density 2^{-32}, leading to immediate break of confidentiality for Robin, iSCREAM, Zorro.

Commuting linear maps in Robin

Robin

Robin and Fantomas [GLSV14], FSE 2014.

Lightweight block ciphers with efficient masking.

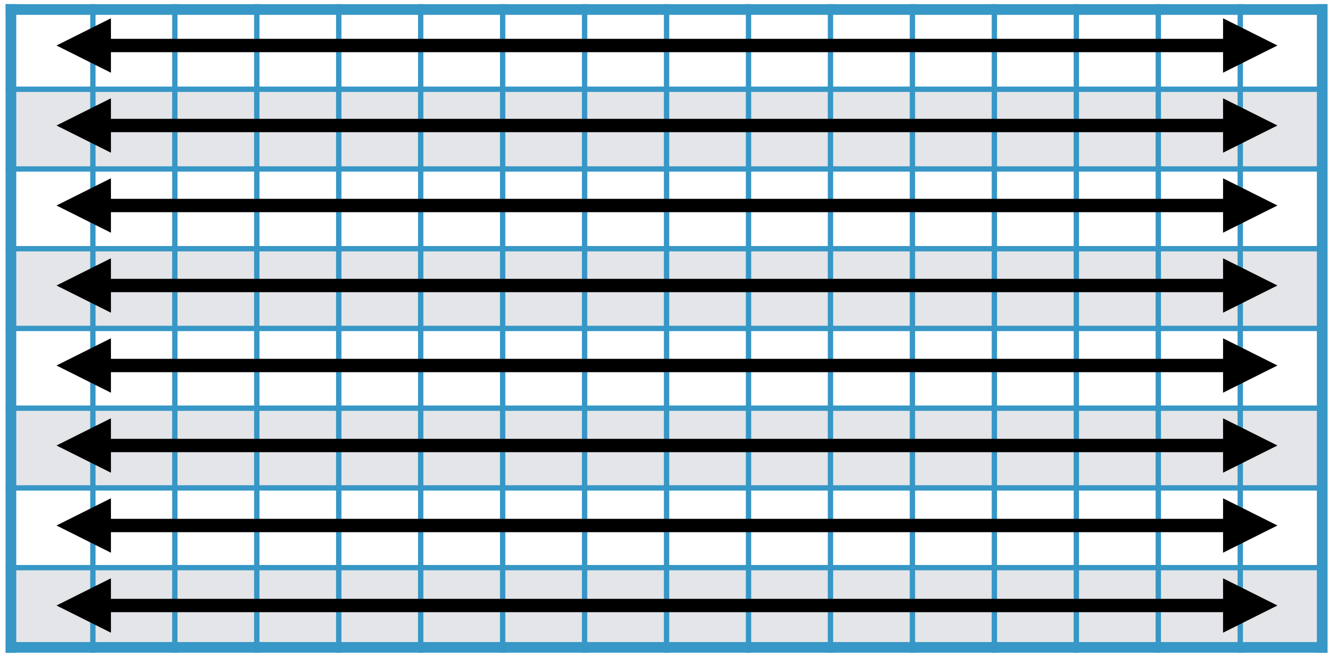
Block =128 bits — Security = 128 bits

Robin = involutive version.

Simple and elegant design: “LS-design”.

Robin: **L** layer

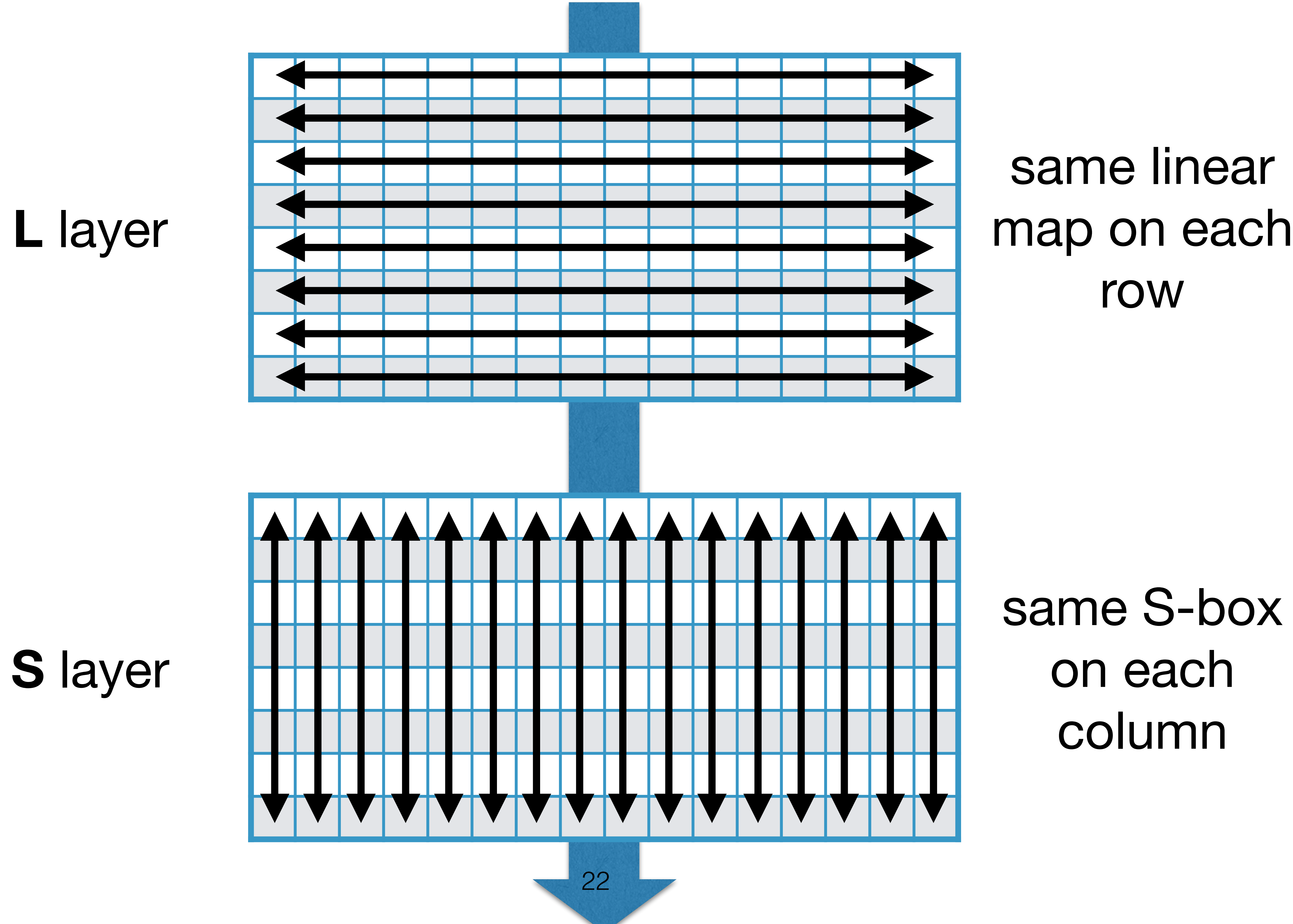
L layer



State
16 x 8 bits

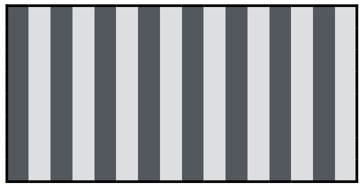
The same linear map *L* is applied to each row.

Robin: **LS** layers



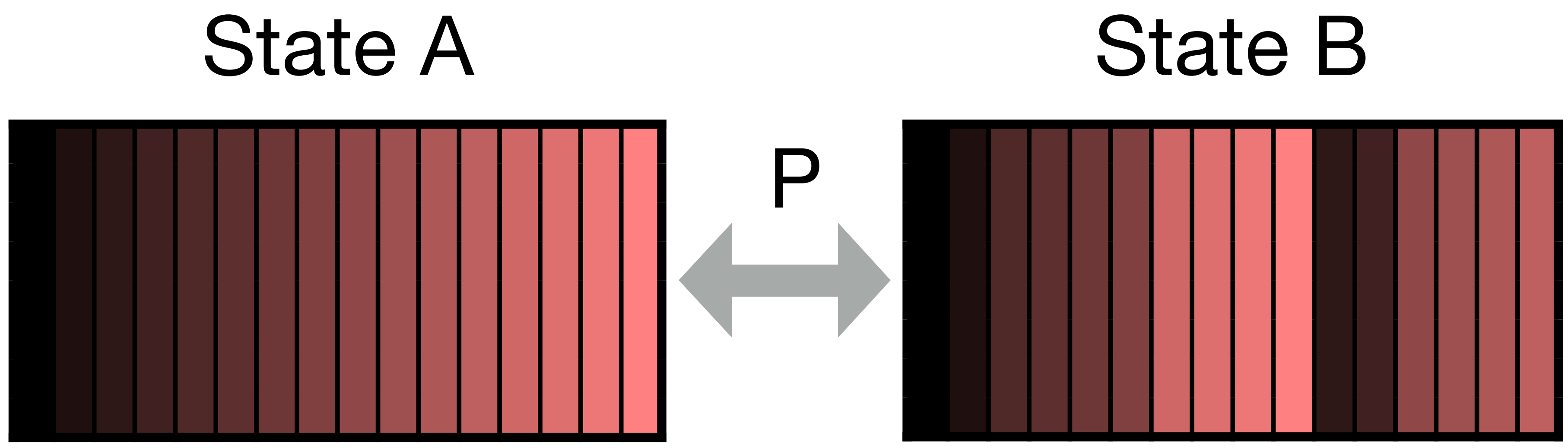
Robin round function

One round =

- L layer 
- S layer 
- Constant addition
- Key addition

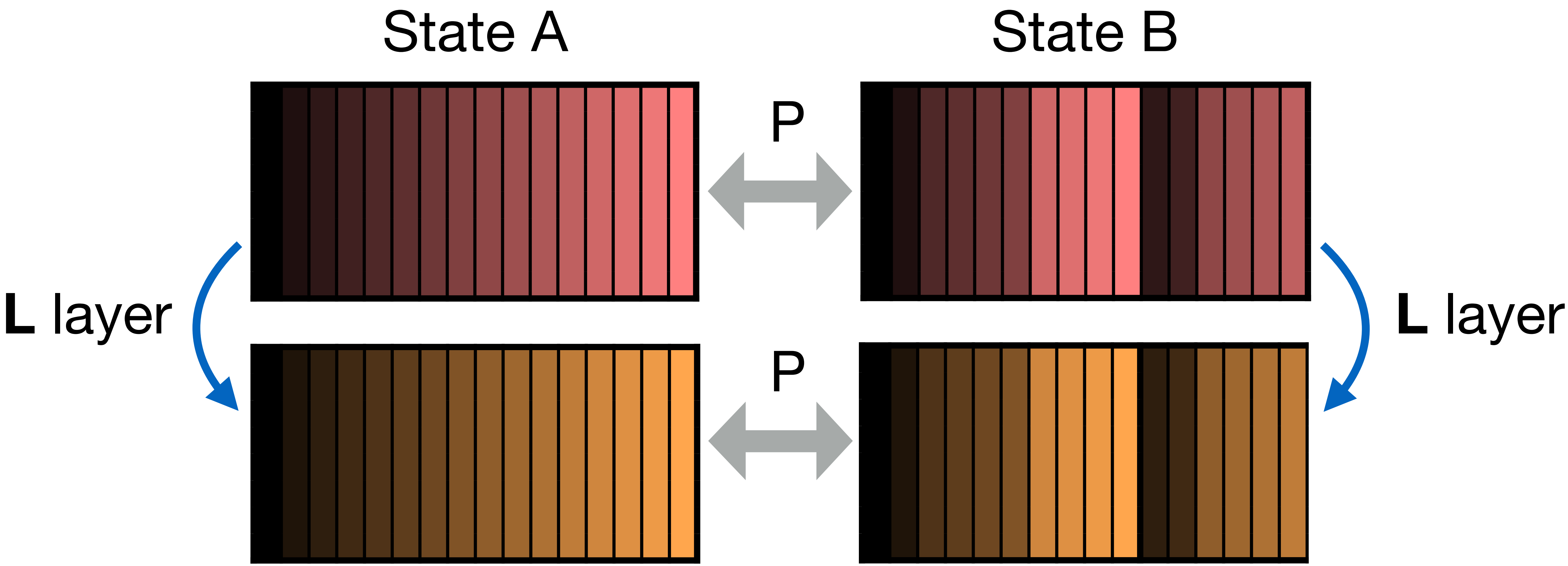
Encryption: 16 rounds.

Invariant permutations



State B = permutation of the columns of state A

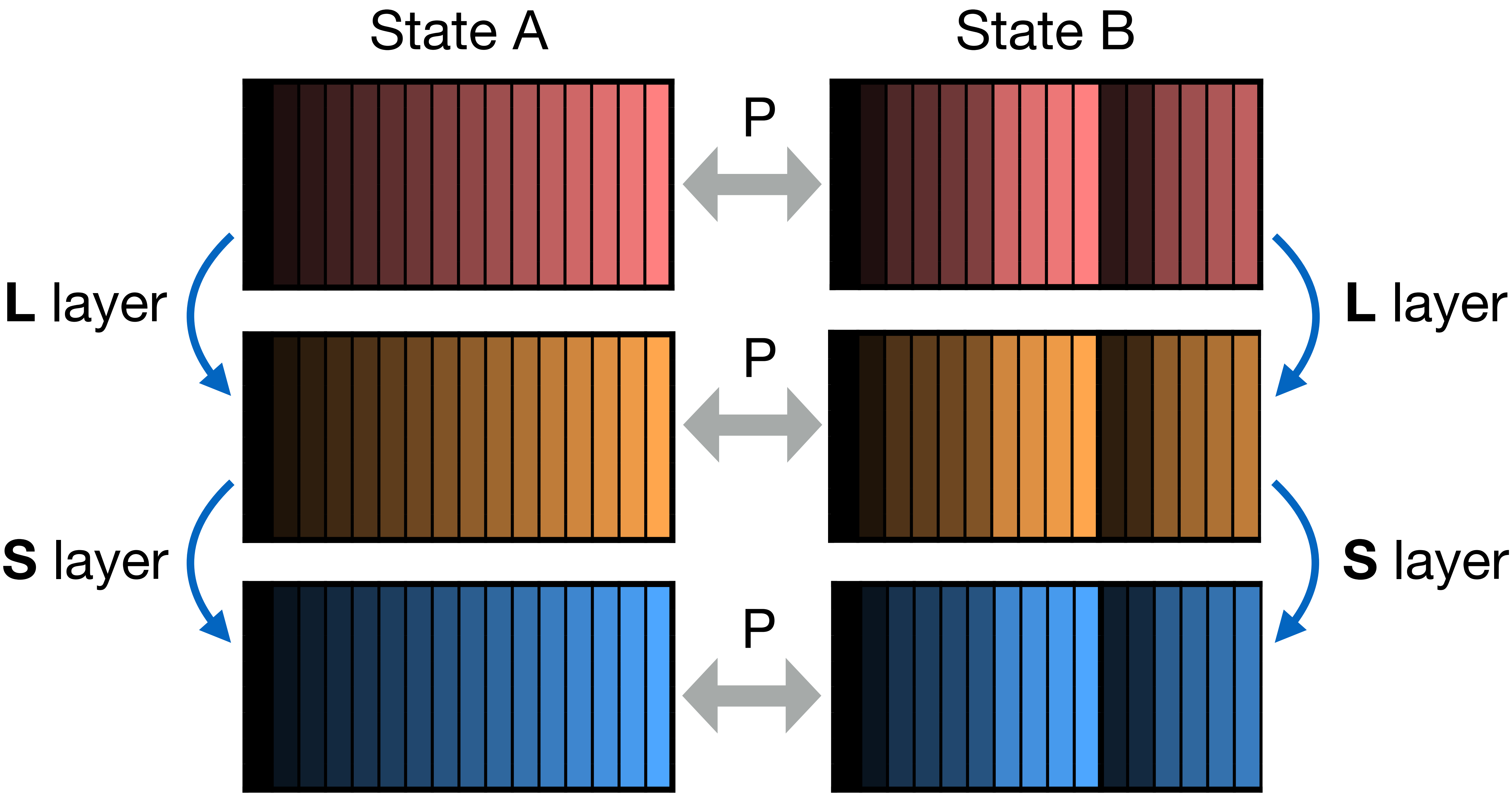
Invariant permutations



Assume **PL = LP**.

Then State B remains a permutation of State A through the **L** layer.

Invariant permutations



The **S** layer comes for free!

Invariant permutations

StateB remains permutation of State A through…

- **L** layer: OK if $LP = PL$.
- **S** layer: OK.
- Constant addition: OK if $P(C_i) = C_i$.
- Key addition: OK if $P(K_A) = K_B$.

➡ P commutes with the round function!

Invariant permutation attack

If $LP = PL$ and $\forall i, C_i \in \ker(P + \mathrm{Id})$:

then for *related keys* $K_2 = P(K_1)$, *related plaintexts* $P_2 = P(P_1)$ remain related through encryption and yield *related ciphertexts* $C_2 = P(C_1)$.

Invariant permutation attack

If $LP = PL$ and $\forall i, C_i \in \ker(P + \mathrm{Id})$:

then for *related keys* $K_2 = P(K_1)$,
related plaintexts $P_2 = P(P_1)$ remain related through encryption and yield *related ciphertexts* $C_2 = P(C_1)$.

If $LP = PL$ and $\forall i, C_i \in \ker(P + \mathrm{Id})$:

then for ***self-related*** *key* $K = P(K)$,
related plaintexts $P_2 = P(P_1)$ remain related through encryption and yield *related ciphertexts* $C_2 = P(C_1)$.

Invariant permutation attack

If $LP = PL$ and $\forall i, C_i \in \ker(P + \mathrm{Id})$:

then for a *self-related* key $K = P(K)$, *self-related* plaintexts $M = P(M)$ yield *self-related* ciphertexts $C = P(C)$.

Invariant permutation attack

If $LP = PL$ and $\forall i, C_i \in \ker(P + \mathrm{Id})$:

then for a *self-related* key $K = P(K)$,
self-related plaintexts $M = P(M)$ yield *self-related* ciphertexts $C = P(C)$.

This is an invariant subspace attack!

The invariant subspace is $\ker(P + \mathrm{Id})$.

Attack on Robin and iSCREAM

Robin and iSCREAM : one suitable permutation P.

- **Weak key** attack. Density $2^{-\operatorname{codim}\ker(P+\mathrm{Id})} = 2^{-32}$
- **Related key** attack.
- Attacks require 2 chosen plaintexts, practically no time or memory.

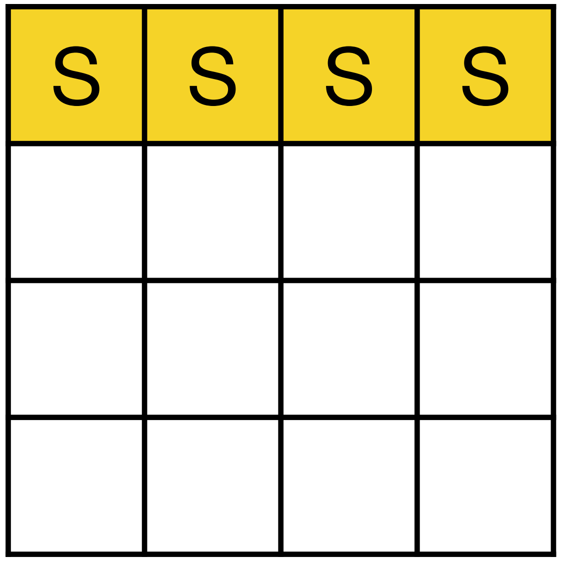
In addition, for weak keys:

- Fixed points of P form a subcipher.
- Key recovery in time 2^{64}.

Robin vs Zorro

Zorro is a variant of AES with some key differences:

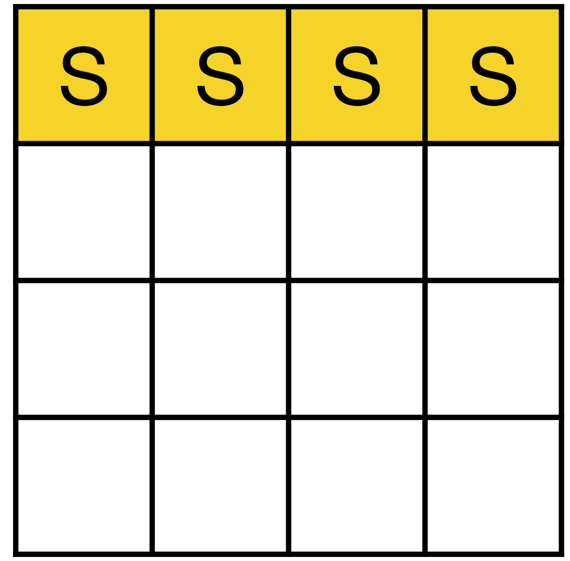
- No key schedule.
- S-boxes affect a single row.



Robin vs Zorro

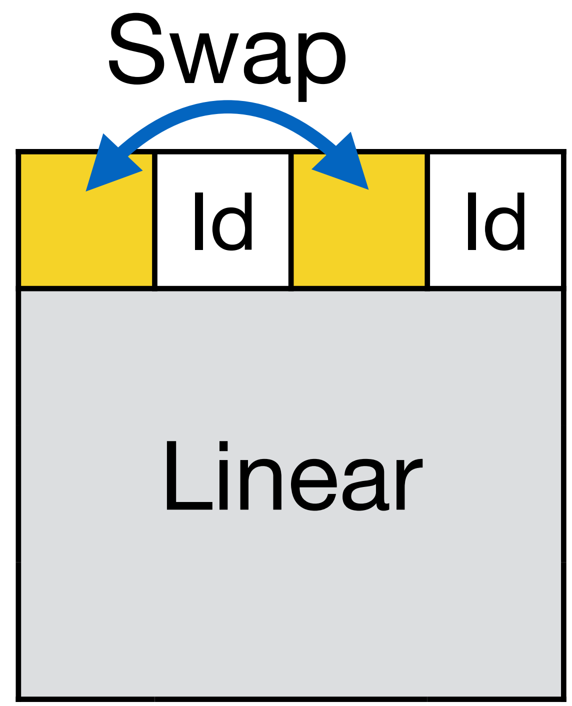
Zorro is a variant of AES with some key differences:

- No key schedule.
- S-boxes affect a single row.



Yet: there still exists *M* that commutes with the round function!

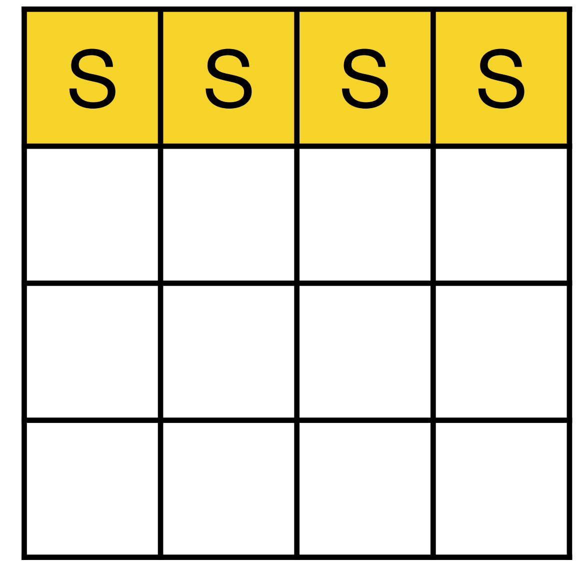
$M =$



Robin vs Zorro

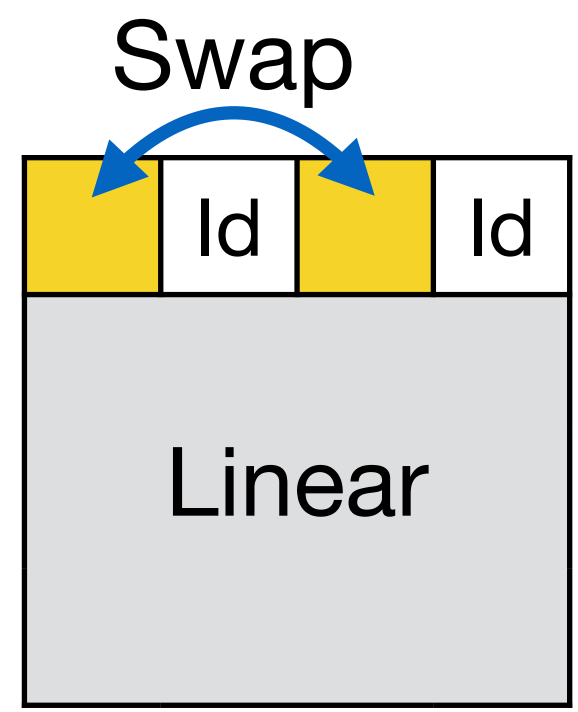
Zorro is a variant of AES with some key differences:

- No key schedule.
- S-boxes affect a single row.



Yet: there still exists M that commutes with the round function!

$M =$



➡ **All** the same weaknesses as Robin.
In particular, weak key set of density 2^{-32}.

Attack comparison

	Type	Data	Time	Reference
Robin, iSCREAM	Weak key, density 2^{-32}	2 CP	negligible	**this paper**
Zorro	Weak key, density 2^{-32}	2 CP	negligible	**this paper**
	Differential	$2^{41.5}$ CP	2^{45}	[BDDLKT14]
	Linear	2^{45} KP	2^{45}	[BDDLKT14]

Conclusion

- A generic algorithm to find invariant subspaces.

  Automatically finds attacks on Robin, iSCREAM and Zorro.

- Practical break of Robin, iSCREAM and Zorro.

  Weak key set of density 2^{-32} in all cases.

  Based on a new self-similarity property.

  Uncovers more properties : commuting linear map, subcipher, faster key recovery…

Conclusion

Thank you for your attention!

Questions ?