# Structural Cryptanalysis of SASAS

Alex Biryukov and Adi Shamir

רז חילו

# Introduction

- Structural cryptanalysis is the branch of cryptology which studies the security of cryptosystems described by generic block diagrams.
- It analyses the syntactic interaction between the various blocks, but ignores their semantic definition as particular functions.
- For Exemple: Meet in in the middle attacks on double encryptions, the study of various chaining structures, and the properties of Feistel structures with a small number of rounds.

# Introduction- Cont.

- Structural attacks are often weaker than actual attacks on given cryptosystems, since they cannot exploit particular weaknesses of concrete functions.
- The flip side of this is that they are applicable to large classes of cryptosystems, including those in which some of the internal functions are unknown or key dependent.
- Structural attacks often lead to deeper theoretical understanding of fundamental constructions, and thus they are very useful in establishing general design rules for strong cryptosystems.

# Introduction- Cont.

- The class of block ciphers considered in this paper are product ciphers which use alternate layers of invertible S-boxes and affine mappings.
- This structure is a generalization of substitution/permutation networks in which the affine mapping is just a bit permutation), and a special case of Shannon's encryption paradigm which mixes complex local operations with simple global operations.

# Introduction- Cont.

- The best example for substitution/affine ciphers is Rijndael which was recently selected as the winner of AES competition.

# Introduction- Cont.

- The best non-structural attack on Rijndael is based on the square attack which exploits the knowledge of the S-box, the simplicity of the key schedule and the relatively slow avalanche of the sparse affine mapping.
- It can break versions with six S-box layers and six affine layers (a seventh layer can be added if the attacker is willing to guess its 128 bit subkey in a nonpractical attack).

# Introduction- Cont.

- In our structural attacks we:
  - Do not know anything about the S-boxes.
  - Do not know the affine mapping.
  - Do not know the key schedule.
- since they can all be defined in a complex key-dependent way.
- We assume that the avalanche is complete after a single layer of an unknown dense affine mapping, and that any attempt to guess even a small fraction of the key would require a nonpractical amount of time.

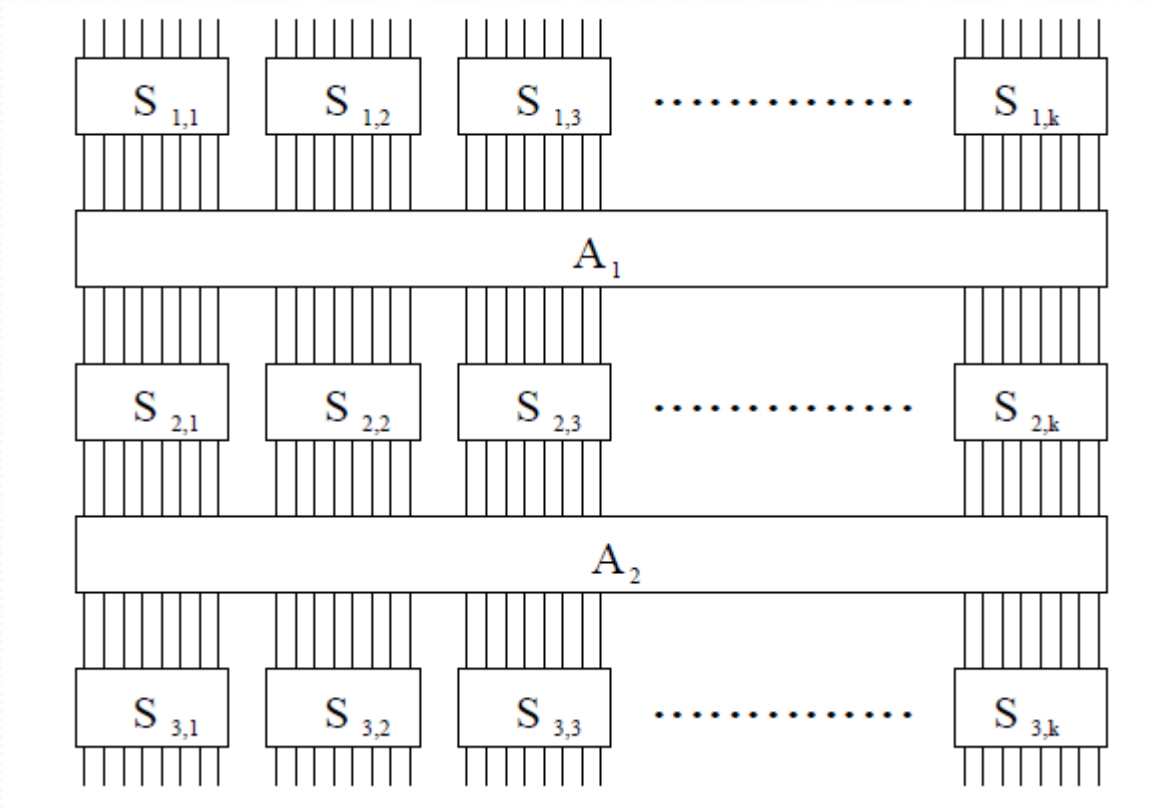
# Introduction- Cont.

- Consequently, we cannot use the square attack and we have to consider a somewhat smaller number of layers.

# Introduction- Cont.

- We describe surprisingly efficient structural attacks on substitution/affine structures with five to seven layers.
- The main scheme we attack is the five layer scheme $S_3 A_2\ S_2\ A_1\ S_1$ in which each S layer contains $k$ invertible S-boxes which map m bits to m bits, and each A layer contains an invertible affine mapping of vectors of n = km bits over GF(2): $A_i(x) = L_i x \oplus B_i$

# Introduction- Cont.



# Introduction- Cont.

- The only information available to the attacker is the fact that the block cipher has this general structure, and the values of k and m.
- Since all the S-boxes and affine mappings are assumed to be different and secret, the effective key length of this five layer scheme is:
- $\log(2^m!)^{3\cdot\frac{n}{m}}+2\log\left(0.29\cdot 2^{n^2}\right)\approx 3\cdot 2^m(m-1.44)\cdot\frac{n}{m}+2n^2$

# Introduction- Cont.

- The new attack is applicable to any choice of m and n, but to simplify the analysis we concentrate on the Rijndael-like parameters of m = 8 bit S-boxes and n = 128 bit plaintexts.
- The effective key length of this version is about:
- $3 \cdot 2^{12} \cdot 6.56 + 2^{15} \approx 113{,}000 \approx 2^{17}$ bits.

# Introduction- Cont.

- It is important to note that not all the information about the S-boxes and the affine mappings can be extracted from the scheme, since there are many equivalent keys which yield the same mapping from plaintexts to ciphertexts.
- Our attack finds an equivalent representation of all the elements in the scheme which makes it possible to encrypt and decrypt arbitrary texts, but it may be different from the original definition of these elements.

# The Multiset Attack- Multiset Properties

- We now develop a calculus of multiset properties, which makes it possible to characterize intermediate values deep in the encryption structure even though nothing is known about the actual functions in it.
- Each multiset can be represented as a list of (value, multiplicity) pairs for example:

  $\{1,1,1,2,2,2,2,7\} \rightarrow (1,3),(2,4),(7,1)$.

- The size of the multiset is the sum of all its multiplicities (in the above example 8).

# The Multiset Attack- Multiset Properties-Cont.

- Definition 1:
  - A multiset M of m-bit values has property C (constant) if it contains an arbitrary number of repetitions of a single value.
  - {1,1,1,2,2,2,2,7}
- Definition 2:
  - A multiset M of m-bit values has property P (permutation) if it contains exactly once each one of the $2^m$ possible values.
  - {0,1,2,3,4,5,6,7}

# The Multiset Attack- Multiset Properties-Cont.

- Definition 3:
  - A multiset M of m-bit values has property E (even) if each value occurs an even number of times (including no occurrences at all).
  - {1,1,2,2,3,3,4,4}
- Definition 4:
  - A multiset M of m-bit values has property B (balanced) if the XOR of all the values (taken with their multiplicities) is the zero vector $0^m$.
  - {1,1,2,2,3,3,4,4,5,5,5,5}

# The Multiset Attack- Multiset Properties-Cont.

- Definition 5:
  - A multiset M of m-bit values has property D (dual) if it has either property P or property E.

# The Multiset Attack- Multiset Properties-Cont.

- Lemma 1:
  - 1. Any multiset with either property E or property P (when m > 1) also has property B.
  - 2. The E and C properties are preserved by arbitrary functions over m-bit values.
  - 3. The P property is preserved by arbitrary bijective functions over m-bit values.
  - 4. The B property is preserved by an arbitrary linear mapping from m bits to n bits when m > 1. It is preserved by arbitrary affine mappings when the size of the multiset is even.

# The Multiset Attack- Multiset Properties-Cont.

- Let us consider now blocks of larger size $n = k \cdot m$ with mixed multiset properties.
- For example, we denote by $C^{i-1}PC^{k-i}$ a multiset with the property that when we decompose each n bit value into k consecutive blocks of m contiguous bits, k-1 of the blocks contain (possibly different) constants across the multiset, and the i-th block contains exactly once each one of the $2^m$ possible m-bit values.

# The Multiset Attack- Multiset Properties-Cont.

- We denote by $D^k$ a multiset that decomposes into k multisets each one of which has property D.
- This decomposition should be understood not as a cross product of k multisets but as a collection of k projections of n bit to m bit values.
- Note that this decomposition operation is usually nonreversible, since we lose the order in which the values in the various blocks are combined.

# The Multiset Attack- Multiset Properties-Cont.

- For example {0,1,2,3}{1,1,2,2}{1,1,1,1} can be derived from several different multisets such as {(011),(111),(221,),(321)} or {(021),(121),(211),(311)}.

# The Multiset Attack- Multiset Properties-Cont.

- Lemma 2:
  - 1. Property $C^{i-1}PC^{k-i}$ is preserved by a layer of arbitrary S-boxes provided that the i-th S-box is bijective.
  - 2. Property $D^k$ is transformed into property $D^k$ by a layer of bijective S-boxes.
  - 3. Property $D^k$ is transformed into $B^k$ by an arbitrary linear mapping on n bits, and by an arbitrary affine mapping when the size of the multiset is even.
  - 4. Property $C^{i-1}PC^{k-i}$ is transformed into property $D^k$ by an arbitrary affine mapping when the size of the multiset is even.

# The Multiset Attack- Multiset Properties-Cont.

- Proof:
  - Claims 1 and 2 are trivial. We show why claim 3 holds.
  - We donate claim 3 as $y_j = \sum_{i=1}^{n} d_{ji} x_i$ a bit $y_j$ at the output of the linear mapping.
  - Property B holds since for each j, the sum (mod 2) of $y_j$ bits over the $2^m$ elements of the multiset is zero:
    - $\sum_{s=1}^{2^m} y_j^s = \sum_{s=1}^{2^m} \sum_{i=1}^{n} d_{ji} x_i^s = \sum_{i=1}^{n} d_{ji} \sum_{s=1}^{2^m} x_i^s = 0$
  - The last expression is zero since by Lemma 1, claim 1, both P and E (and thus D) imply the B-property.

# The Multiset Attack- Multiset Properties-Cont.

- The result remains true even when we replace the linear mapping by an affine mapping if we XOR the additive constant an even number of times.
- Let us now show why claim 4 holds:
- Any affine mapping over GF(2) can be divided into k distinct n to m-bit projections.

# The Multiset Attack- Multiset Properties-Cont.

- Since $(k-1)m$ of the input bits are constant, we will be interested only in restrictions of these affine mappings to new affine mappings that map the $i-th$ block of m bits (the one which has the P property) into some other m-bit block in the output:
  - $y = A_{ij}(x) = L_{ij} \cdot x \oplus B_j, j = 1, \ldots, k$
- Here $L_{ij}$ is an arbitrary $m \times m$ (not necessarily invertible) binary matrix and $B_j \in \{0,1\}^m$.
- We can again ignore $B_j$ since it is XOR'ed an even number of times.

# The Multiset Attack- Multiset Properties-Cont.

- If $L_{ij}$ is invertible over GF(2), then $L_{ij} \cdot x$ is a 1-1 transform and thus $L_{ij} \cdot x$ gets all the $2^m$ possible values when x ranges over all the $2^m$ possible inputs, so it has property P.
- Thus we are left with the case of non-invertible $L_{ij}$.
- Suppose that $rank(L_{ij}) = r < m$.
- The kernel is defined as the set of solutions of the homogeneous linear equation $L_{ij} \cdot x = 0$.

# The Multiset Attack- Multiset Properties-Cont.

- Let $x_0$ be some solution of the non-homogeneous equation $L_{ij} \cdot x = y$.
- Then all the solutions of the non-homogeneous equation have the form $x_0 \oplus v_0$, where $v_0$ is any vector from the kernel.
- The size of the kernel is $2^{m-r}$, and thus each y has either no preimages or exactly $2^{m-r}$ preimages.

# The Multiset Attack- Multiset Properties-Cont.

- Since $r < m$ by assumption, $2^{m-r}$ is even, and thus the multiset of m-bit results has property E.
- Consequently each block of m bits of the output has either property P or property E, and thus the n bit output has property $D^k$, as claimed.

# The Multiset Attack- Recovering Layers $S_1$ and $S_3$

- Consider a multiset of chosen plaintexts with property $C^{i-1}PC^{k-i}$.
- The key observations behind the attack are:
  - 1. The given multiset is transformed by layer $S_1$ into a multiset with property $C^{i-1}PC^{k-i}$ by Lemma 2, claim 1.
  - 2. The multiset $C^{i-1}PC^{k-i}$ is transformed by the affine mapping $A_1$ into a multiset with property $D^k$ by Lemma 2, claim 4.

# The Multiset Attack- Recovering Layers $S_1$ and $S_3$-Cont.

- 3. The multiset property $D^k$ is preserved by layer S2, and thus the output multiset is also $D^k$, by Lemma 2, claim 2.
- 4. The multiset property $D^k$ is not necessarily preserved by the affine mapping $A_2$, but the weaker property $B^k$ is preserved.
- 5. We can now express the fact that the collection of inputs to each S-box in $S_3$ satisfies property B by a homogeneous linear equation.
- We will operate with m-bit quantities at once as if working over GF($2^m$) (XOR and ADD are the same in this field).

# The Multiset Attack- Recovering Layers $S_1$ and $S_3$-Cont.

- Variable $z_i$ represents the m-bit input to the S-box which produces i as an output (the variable $z_i$ describes $S^{-1}$, which is well defined since S is invertible), and we use $2^m$ separate variables for each S-box in $S_3$.
- When we are given a collection of actual ciphertexts, we can use their m-bit projections as indices to the variables, and equate the XOR of the indexed variables to $0^m$.
- Different collections of chosen plaintexts are likely to generate linear equations with different random looking subsets of variables.

# The Multiset Attack- Recovering Layers $S_1$ and $S_3$-Cont.

- When sufficiently many linear equations are obtained we can solve the system by Gaussian elimination in order to recover all the S-boxes in $S_3$ in parallel.

# The Multiset Attack- Recovering Layers $S_1$ and $S_3$-Cont.

- Unfortunately, we cannot get a system of equations with a full rank of $2^m$, Consider the truth table of the inverted S-box as a $2^m \times$ m-bit matrix.
- Since the S-box is bijective, the columns of this matrix are m linearly independent $2^m$-bit vectors.
- Any linear combination of the S-box input bits (which are outputs of the inverted S-box) is also a possible solution, and thus the solution space must have a dimension of at least m.

# The Multiset Attack- Recovering Layers $S_1$ and $S_3$-Cont.

- Moreover, since all our equations are XOR's of an even number ($2^m$) of variables, the bit complement of any solution is also a solution.
- Since the system of linear equations has a kernel of dimension at least m+1, there are at most $2^m - m - 1$ linearly independent equations in our system.
- When we used m=8 we got a linear system of rank 247 in 256 variables, according to the formula.

# The Multiset Attack- Recovering Layers $S_1$ and $S_3$-Cont.

- Fortunately, this rank deficiency is not a problem in our attack. When we pick any one of the non-zero solutions, we do not get the "true" $S^{-1}$, but A($S^{-1}$), where A is an arbitrary invertible affine mapping over m-bits.
- By taking the inverse we obtain S($A^{-1}$).

# The Multiset Attack- Recovering Layers $S_1$ and $S_3$-Cont.

- This is the best we can hope for at this phase, since the arbitrarily chosen $A^{-1}$ can be compensated for when we find $A(A_2) = A'_2$ instead of the "true" affine transform A2, and thus the various solutions are simply equivalent keys which represent the same plaintext/ciphertext mapping.

# The Multiset Attack- Recovering Layers $S_1$ and $S_3$-Cont.

- single collection of $2^m$ chosen plaintexts gives rise to one linear equation in the $2^m$ unknowns in each one of the $k$ ·S-boxes in layer $S_3$.
- To get $2^m$ equations, we can use $2^{2m}$ $(2^{16})$ chosen plaintexts of the form $(A, u, B, v, C)$, in which we place the P structures u and v at any two block locations, and choose A,B,C as arbitrary constants.
- For each fixed value of u, we get a single equation by varying v through all the possible $2^m$ values.
- However, we can get an additional equation by fixing v and varying u through all the $2^m$ possible values.

# The Multiset Attack- Recovering Layers $S_1$ and $S_3$-Cont.

- Since we get $2 \cdot 2^m$ equations in $2^m$ unknowns, we can reduce the number of chosen plaintexts to $\frac{3}{4} \cdot 2^{2m}$ by eliminating the $\frac{1}{4}$ of the plaintexts in which u and v are simultaneously chosen in the top half of their range.

# The Multiset Attack- Recovering Layers $S_1$ and $S_3$– Cont.

- Solving each system of linear equations by Gaussian elimination requires $2^{3m}$ steps, and thus we need $k \cdot 2^{3m}$ steps to find all the S-boxes in $S_3$.
- For n=128, m=8, k=16 we get very modest time complexity of $2^{28}$.
- To find the other external layer $S_1$, we can use the same attack in the reverse direction. However, the resultant attack requires both chosen plaintexts and chosen ciphertexts.

# The Multiset Attack- Attacking the Inner Layers ASA

- We are left with a structure $A'_2\ S_2\ A'_1$ - two (possibly modified) affine layers and an S-box layer in the middle.
- In order to recover the affine layers we use Biham's low rank detection technique.
- Consider an arbitrary pair of known plaintexts $P_1$ and $P_2$ with difference $P_1 \oplus P_2$.
- With probability $\frac{k}{2^m}$, after $A'_1$ there will be no difference at the input to one of the k S-boxes in $S_2$. Thus there will also be no difference at the output of this S-box.

# The Multiset Attack- Attacking the Inner Layers ASA-Cont.

- Consider now the set of pairs $P_1 \oplus C_i$, $P_2 \oplus C_i$ for many randomly chosen n-bit constants $C_i$.
- Any pair in this set still has this property, and thus the set of all the obtained output differences after $A'_2$ will have a rank of at most $n - m$, which is highly unusual for random n dimensional vectors.
- Consequently, we can confirm the desired property of the original pair $P_1$ and $P_2$ by applying this low rank test with about n modifiers $C_i$.

# The Multiset Attack- Attacking the Inner Layers ASA-Cont.

- We want to generate and test pairs with zero input differences at each one of the k S-boxes.
- We choose a pool of t random vectors $P_j$ and another pool of n modifiers $C_i$, and encrypt all the $n \cdot t$ combinations $P_j \oplus C_i$.
- We have about $\frac{t^2}{2}$ possible pairs of $P_j$ 's, each one of them has a probability of $\frac{k}{2^m}$ to have the desired property at one of the S-boxes, and we need about $k \cdot \log(k)$ random successes to cover all the k S-boxes.

# The Multiset Attack- Attacking the Inner Layers ASA-Cont.

- The critical value of t thus satisfies $\frac{t^2}{2} \cdot \frac{k}{2^m} = k \cdot \log(k)$ and thus $t = \sqrt{2^{m+1}\log(k)}$.
- For n=128, m=8, k=16 we get $t=2^{5.5}$, and thus the total number of chosen plaintexts we need is $n \cdot t = 2^{12.5}$.

# The Multiset Attack- Attacking the Inner Layers ASA-Cont.

- Now we use linear algebra in order to find the structure of $A'_2$.
- Consider the representation of $A'_2$ as a set of n vectors $V_0, V_1, \ldots, V_{n-1}, V_i \in \{0,1\}^n$, where $A'_2$ transforms an arbitrary binary vector $b_0, b_1, \ldots, b_{n-1}$ producing the linear combination:
  - $A'_2(\mathrm{b}) = \bigoplus_{i=0}^{n-1} b_i V_i$.

# The Multiset Attack- Attacking the Inner Layers ASA-Cont.

- From the data pool we extract information about k different linear subspaces of dimension $n - m$ $(= 120)$.
- Then we calculate the intersection of any $k - 1 (= 15)$ of them.
- This intersection is an m-dimensional linear subspace which is generated by all the possible outputs from one of the S-boxes in layer $S_2$, after it is expanded from 8 bits to 128 bits by $A'_2$.

# The Multiset Attack- Attacking the Inner Layers ASA-Cont.

- We perform this operation for each S-box and by this we find a linear mapping $A^*_2$ which is equivalent to the original choice.
- The complexity of this phase is that of Gaussian elimination on a set of $O(n-m)$ equations.

# The Multiset Attack- Attacking the Inner Layers ASA-Cont.

- After finding and discarding $A'_2$, we are left with the two layer structure $S_2\ A'_1$.
- If we need to perform only decryption, we can recover this combined mapping by writing formal expressions for each bit, and then solving the linear equations with $k \cdot 2^m$ $(2^{12})$ variables.
- If we also need to perform encryption this trick will not work, since the formal expressions will be huge, However, we can just repeat our attack in the reverse direction by using chosen ciphertexts and recover $A^*_1$.

# The Multiset Attack- Attacking the Inner Layers ASA-Cont.

- After that we can find the remaining layer $S_2$ with about $2^m$ known plaintexts.
- Again we will find not the real S-box layer $S_2$ but the equivalent one which corresponds to the modified $A^*_1$, $A^*_2$ that we have found in earlier phases.

# The Multiset Attack- Attacking the Inner Layers ASA-Cont.

- The complete attack uses about $2^{2m}$ chosen plaintexts $(2^{16})$ and about $k2^{3m}$ $(16 \cdot 2^{24} = 2^{28})$ steps.

# A Chosen Plaintext Attack on ASAS

- We now show how to use a pure chosen plaintext attack, and avoid the less realistic chosen plaintext and chosen ciphertext attack with the same time and data complexities as the original attack.
- After the first phase of the original attack we are left with a $A'_2 S_2 A_1 S_1$ structure, since we can recover only one of the two external S-box layers.

# A Chosen Plaintext Attack on ASAS- Cont.

- Since the inputs go through the additional S-box layer $S_1$, we can no longer argue that for any $C_i$, $P_1 \oplus C_i$ and $P_2 \oplus C_i$ will have a zero difference at the input to some S-box in $S_2$ whenever $P_1$ and $P_2$ have this property.
- Because of that we have to use a more structured set of modifiers which can be nonzero only at the inputs to the S-boxes in which $P_1$ and $P_2$ are identical.

# A Chosen Plaintext Attack on ASAS- Cont.

- We use $2^{16}$ chosen plaintexts with the multiset property $PPC^{k-2}$ (the two P's could be placed anywhere, and we could reuse the chosen plaintexts from the first phase of the attack).
- There are $2^{15}$ different ways to choose a pair of values from the first P.
- For each such pair $(a_1, a_2)$, we generate a group of $2^8$ pairs of extensions of the form $(a_1, b_0, c, d, \dots)$ and $(a_2, b_0, c, d, \dots)$ where $b_0$ is any common element from the second $P$, and $c, d, \dots$ are the constants from $C^{k-2}$.

# A Chosen Plaintext Attack on ASAS- Cont.

- We claim that all these $2^8$ pairs will have the same difference at the output of $S_1$, since the first S-box gets a fixed pair of values and the other S-boxes get identical inputs in each pair.
- We can now apply the low rank test since we have sufficiently many choices of $(a_1, a_2)$ to get a zero difference at the input to each S-box in $S_2$ with high probability, and for any such $(a_1, a_2)$ we have sufficiently many pairs with the same difference in order to reliably test the rank of the output vectors.

# A Chosen Plaintext Attack on ASAS- Cont.

- Once we discover the partition of the output space into 16 different linear subspaces of dimension 120, we can again find the intersection of any 15 of them in order to find the 8 dimensional subspace generated by the outputs of each one of the 16 S-boxes.
- We fix $A'_2$ by choosing any set of 8 arbitrary spanning vectors in each one of the 16 subspaces, and this is the best we can possibly do in order to characterize $A'_2$ due to the existence of equivalent keys.

# A Chosen Plaintext Attack on ASAS- Cont.

- One possible problem with this compact collection of plaintexts is that the attack may fail for certain degenerate choices of affine mappings.
- For example, if both $A_1$ and $A_2$ are the identity mapping, the insufficiently mixed intermediate values always lead to very low output ranks.
- However, the attack was always successful when tested with randomly chosen affine mappings.

# A Chosen Plaintext Attack on ASAS- Cont.

- After peeling off the computed $A'_2$, we are now left with a $S'_2 A_1 S_1$ structure, which is different from the $A'_2 S_2 A'_1$ structure we faced in the original attack.
- We have already discovered in the previous part of the attack many groups of 256 pairs of plaintexts, where in each group we know that the XOR of each pair of inputs to any particular S-box in $S'_2$ is the same constant.

# A Chosen Plaintext Attack on ASAS- Cont.

- We do not know the value of this constant, but we can express this property as a chain of homogeneous linear equations in terms of the values of the inverse S-box, which are indexed by the known outputs from the $S'_2 A_1 S_1$ structure.
- For example:
  - $S^{-1}(1) \oplus S^{-1}(72) = S^{-1}(255) \oplus S^{-1}(13) = S^{-1}(167) \oplus S^{-1}(217)$=…

# A Chosen Plaintext Attack on ASAS- Cont.

- If we need additional equations, we simply use another one of the $2^{15}$ possible groups of pairs, which yields a different chain of equations (with a different unknown constant).

Questions?