The CSQUARE Transform[⋆]

Tom St Denis

The A.F.A.H.S.G. Institute
tomstdenis@iahu.ca

Abstract. In this paper we show how to combine the design concepts of the SQUARE and CS block ciphers to produce a pseudo-random permutation CSQUARE suitable for use in block cipher and hash design with a very high multi-round trail weight. The new design inherits the hardware efficiency of the SQUARE linear transform pattern as well as the efficiency of the fast pseudo-Hadamard transform over a finite field. We demonstrate the DMWT hash function which makes use of our new results.

Keywords. Pseudo-Hadamard Transform, Branch Analysis, Pseudo-Random Permutations.

1 Introduction

In the SQUARE block cipher [1] a new technique of combining a Maximum Distance Separable (MDS) code along with the "SQUARE Propagation Pattern" resulted in a design which had a high provable four round differential and linear trail weight. Similarly in the CS-Cipher [3] the concept of a pseudo-Hadamard transform was enhanced to raise the minimal eight round differential and linear trail weight.

In this paper we take the results of FPHT analysis in [7] and apply them to the SQUARE propagation pattern to derive a new permutation which has a considerably higher minimal trail weight. We prove that through our DMWT design any four round differential or linear trail has a weight of at least 144 which compares favourably to the trail weight of WHIRLPOOL which would be at least 81.

Along with our successful design we also demonstrate two similar design constructions which fail to achieve a trail weight greater than the trail weight of the SQUARE design. In section two we discuss the relevant background required for our new results. Section three discusses our new design as well as the two similar designs. We shall conclude this paper with a section discussing implementation details and our DMWT hash function.

⋆ Note this paper is a work in progress. Draft January 30th, 2004

2 Background Theory

2.1 SQUARE

The SQUARE [1] block cipher was a unique approach to cipher design when it was first proposed. It was an efficient design that provided for a high and provable minimal trail weight. The design would later change into the Rijndael block cipher which became the American AES cipher.

The relevant design concepts of SQUARE are the following. Let θ represent a transform which applies a non-linear substitution to all of the coordinates of the input. Let ζ represent a length four MDS code over $GF(2)[x]$. Let γ represent the sixteen coordinate input organized as a 4×4 matrix. Then the relevant portions of the SQUARE round function can be expressed as the following.

$$\gamma_{i+1} = \rho(\gamma_i, K_i) : \gamma_i \times K_i \rightarrow (\zeta \circ \theta \circ (\gamma_i \oplus K_i))^T \tag{1}$$

While it is not proven in the SQUARE paper the design has two provable qualities which are proven in the subsequent Rijndael papers. Let β represent the branch of the ζ transform.

Theorem. Every two rounds has at least β active columns.
Theorem. Every four rounds has at least β^2 active coordinates.

For the purposes of this paper we are merely concerned with the first theorem. Which is trivial to prove. If there are x active columns then through the transposition there shall be one row with at least x active coordinates. Through the subsequent ζ transform there shall be the greatest of $(\beta - x, 1)$ active coordinates. As a result there are $x + \beta - x = \beta$ active columns.

2.2 CS-Cipher

The CS-Cipher [3] used non-linear multipermutations interconnected in a fashion to promote fast diffusion. Instead of placing the non-linear substitutions at the boundaries of the linear transform they placed them throughout the transform. In [4] the full eight round trail weight was counted by brute force to prove the security of the design against differential and linear cryptanalysis. This design in particular has several useful properties.

1. The round function can be computed in $O(\log n)$ time.
2. The mixing network is non-linear.
3. The branch of the round function is bounded.

We use all three properties in our new design by generalizing both of the CS-Cipher and SQUARE designs.

3 New Design - CSQUARE

3.1 Construction

Our new design permutes a block of data γ organized as a square matrix by combining a CS-Cipher style mixing network with the SQUARE propagation pattern. Let $n = 2^k$ for $k \geq 2$ represent the width of γ. Let ζ represent the mixing network made of a H_k transform implemented as [7, Fig.1] with the H_1 transforms replaced with a non-linear $(2, 3)$-multipermutation θ [6]. Let β_k represent the branch of the ζ transform.

We note that with each application of θ within ζ the inputs are "keyed" by some key K. This allows a Markov chain to be produced throughout the design.

Our new design is simply stated as

$$y = \rho(\gamma, K) : \gamma \times K \to \left(\zeta \circ (\zeta \circ \zeta \circ \gamma)^T\right)^T \tag{2}$$

Note that the three compositions of ζ require $3k \cdot 2^{2k}$ key words to be provided by some form of key schedule. For the purposes of this paper we shall assume some form of suitable key schedule has been provided. This design relies on two theorems to achieve a provable four round trail weight bound.

3.2 Design Theory

Theorem 1. *Two compositions of ρ will always have at least β_k active columns.*

Proof. This proof is simply an adaptation of the two round proof for the SQUARE block cipher. If the input has x active columns then the transposition will have one column with at most x active coordinates. Through the third ζ composition of the ρ transform there shall be the largest of $(\beta_k - x, 1)$ active coordinates. When the matrix is transposed again those active coordinates are placed in unique columns. $\square$

The second theorem we are concerned with is concretely establishing the two round trail weight of ζ.

Theorem 2. *The two round trail weight for a ζ transform of dimension $n = 2^k$ is at least $3 \cdot 2^{k-1}$.*

A general proof of this theorem is still being developed. However, we can show that the theorem is true for $k = \{1, 2, 3, 4\}$ and we shall provide an argument in favour of the theorem.

Case $k = 1$. This case is fairly trivial. The branch of θ is three which immediately yields a minimum two round weight of three.

Corollary 3. *Over two layers of ζ any active θ transform will be directly related to at least three active coordinates.*

Case $k = 2$. In both rounds at least one θ transform must be active as a result at least six active coordinates must be present. Therefore, the theorem is true for $k = 2$ since $3 \cdot 2^{k-1} = 6$.

Case $k = 3$. In this case we rely on the branch of ζ_3 which is provably [7] six. As a result the minimal trail weight is caused an input weight of two which cause a minimal output weight of four after the first round. If the layers of the two rounds are placed end-to-end it is obvious that over the first two layers there is at least one active θ. Over the middle two rounds there is at least two and over the last two at least one more active θ. Therefore, there are four active θ transforms which produce the $3 \cdot 2^{k-1} = 12$ active coordinates.

Case $k = 4$. This proof is similar to the case of $k = 3$. Again the lowest weight trail is caused by the input weight of two. For the first round the first two layers will have at least one active θ and the last two layers will have at least three active θ transforms. The second round at the very least will be a mirror of the first round. In total there shall be at least eight active θ transforms which achieves the desired $3 \cdot 2^{k-1} = 24$ active sboxes.

Conjecture for $k \geq 5$. Our conjecture for the general case lies in the topology of the underlying mixing network. In every layer each of the θ transforms are applied to a unique pairing of the input coordinates. As a result in layer i of a transform any active θ transform will be connected to at least two active θ transforms in at least one of layer $i-1$ or $i+1$. It seems very plausible that over the k pairs of layers in the two rounds of ζ_k transforms there are at least 2^{k-1} active θ transforms.

3.3 Results

As a result of the two theorems the two round trail weight of the ρ transform is at least the product of the branch of ζ_k and the two round trail weight of ζ_k. Let σ_n denote the four round trail weight of the ρ construction with a dimension of $n \times n$.

$$\begin{aligned} \sigma_n &= 6 \cdot 8^k, \text{ if } n = 2^{2k} \\ \sigma_n &= 18 \cdot 8^k, \text{ if } n = 2^{2k+1} \end{aligned} \tag{3}$$

The four round minimal trail weight of the SQUARE design is the branch of the "MixColumn" transform squared. The trail weights for even dimension ζ match the SQUARE trail weights at approximately the point $k = 2.541814439$ and at the point $k = 2.169925002$ for odd dimension ζ. As a result the new design achieves a lower four round trail weight for all dimension greater than 32×32.

Dimension	Our Design: ρ	Square Design
4×4	48	25
8×8	144	81
16×16	384	289
32×32	1152	1089
64×64	3072	4225

Fig. 1. Four round minimal trail weights of SQUARE and ρ

3.4 Related Designs

Through the course of this study two other design constructs have been analyzed and both fail to achieve the desired level of trail weight. The first design construct is based on the SQUARE design mixed with the CS-Cipher directly. That is,

$$y = \hat{\rho}_1(\gamma, K) : \gamma \times K \to (\zeta \circ \gamma)^T \tag{4}$$

However, this design lacks[2] an easy method of counting the trail weight over multiple rounds. The second design was a straight adaptation of this design.

$$y = \hat{\rho}_2(\gamma, K) : \gamma \times K \to (\zeta \circ \zeta \circ \gamma)^T \tag{5}$$

However, this design does not enforce a high enough active column count over two rounds. Another modification we considered but rejected for efficiency reasons is the construction:

$$y = \hat{\rho}_3(\gamma, K) : \gamma \times K \to \left(\hat{\zeta} \circ (\zeta \circ \zeta \circ \gamma)^T\right)^T \tag{6}$$

Where $\hat{\zeta}$ is a 2^k dimension MDS code. The resulting four round minimal trail weight of this design would be

$$\hat{\sigma}_n = 3 \cdot \left(4^k + 2^k\right), \text{ if } n = 2^k \tag{7}$$

Which compares very well against the original SQUARE design. However, the $\hat{\rho}_3$ approach involves using an MDS code which is something we are striving to avoid since they are slower than the ζ transform to implement. Therefore, we rejected this approach.

4 Design Justification

4.1 Security

The purpose of this new design is to modify the original SQUARE design and produce a transform which has a higher multi-round trail weight. In doing so we

[2] As of the time of this writing.

Dimension	**Our Design: $\hat{\rho}_3$**	**Square Design**
4×4	60	25
8×8	216	81
16×16	816	289
32×32	3168	1089
64×64	12480	4225

Fig. 2. Four round minimal trail weights of SQUARE and $\hat{\rho}_3$.

also removed much of the linearity from within the design using the approach of [3]. As a result differential and linear cryptanalysis are easy to prove ineffective very early in a multi-round design that uses the ρ transform.

We have also applied a summation attack against a 8×8 implementation of the ρ function. Using single and double coordinate input counters[3] we have confirmed that the design does not emit any useful characteristics over two rounds and that three rounds seems to resist the attack.

4.2 Hardware Efficiency

The first property of this design that is immediately obvious is that it requires more transformations than the SQUARE design. The ρ function applies three transformations per round compared to the single transformation used in a SQUARE round.

However, the saving grace of this design is that the ζ transform has an efficient $O(n)$ space and $O(\log n)$ time implementation. In fact the serious bottleneck of this design is the non-linear transform applied within θ. The actual transform that produces the $(2,3)-$multipermutation quality desired can be accomplished with a very small amount of EOR gates. A further optimization is that since a single ζ transform is likely to be very small[4] it can be implemented in parallel for every column of the input. As a result the critical path of the transform is at least as long as three invocations of the ζ transform.

While no concrete implementation of this design exists it is not likely to be much slower than the original SQUARE design for small dimensions and would be much faster for large dimensions.

4.3 Software

In software this design would be much slower than the SQUARE approach for small dimensions as table driven approaches cannot be used as effectively. However, the approach is not unreasonable and can be implemented using a very small amount of code and data memory[5].

[3] Over all 2^8 and 2^{16} values respectively.
[4] Given a reasonable dimension.
[5] Comparable to an implementation of the SQUARE design for equal dimensions.

5 Example Construction

We have taken the 8×8 ρ function and produced a $512-$bit message digest function named DMWT. Each coordinate of γ is an eight bit value. The non-linear multipermutation θ is constructed by applying the substitution function Σ from WHIRLPOOL[6] [2] followed by the H_1 FPHT over $GF(2)[x]/(x^8 + x^4 + x^3 + x + 1)$.

To make the design simpler to implement we break convention and apply the ζ transform across a row since the data is loaded in row-major order. Each message is padded with a single 1 bit followed by enough zero bits to make the message length congruent to 448 modulo 512 bits. The original message length is stored in little endian fashion as a 64-bit integer at the end of the padding.

The $512-$bit state is organized as 64 eight bit words $S_{0..63}$ and is initially set to the first 64 values of Σ. The message is compressed in blocks of 64 eight bit words $M_{0..63}$. The compression function is created by applying a cipher in Davies-Meyers [5] mode.

```
1. $M' \leftarrow M, S' \leftarrow S$
2. for $x = 0$ to 8 do
    (a) $S' \leftarrow \rho(S', M')$
    (b) $M' \leftarrow \rho(M', \Sigma_{24x})$
3. $S' \leftarrow \rho(S', M')$
4. $S \leftarrow S \oplus S'$
```

In step (2.a) the notation Σ_{24x} implies using the values of $\Sigma_{24x...24x+63}$ as the key for the ρ function. The key schedule for the ρ function is equally simple. We use the words of the 64 element key cyclically with the convention that ζ is applied to the zeroth row first and the ζ network is computed in three layers from top to bottom, left to right with the zero'th coordinate to the left.

5.1 Efficiency

Since this design could not be very fast the design was optimized to save space. By using the Σ contents as the initial state and the keys for the key schedule 640 bytes of code space were saved. With GCC 3.3.2 optimizing for space on the AMD Athlon processor the entire hash occupies $1,836$ bytes[7].

5.2 Security

This design uses ten rounds of the ρ function which means that over eight rounds there are at least $2 \cdot 18 \cdot 8 = 288$ active sboxes. We recall from [2] that the Σ was chosen to have a differential and linear profile maximum of 2^{-5} and 2^{-3}

[6] We used the revised edition of the WHIRLPOOL design.

[7] Which compares nicely to MD2 which with the same compiler options occupies $1,484$ bytes.

respectively. As a result the probability and deviation of the best eight round differential and linear attacks would be 2^{-1440} and 2^{-577} respectively.

The saturation attack was also applied to this design and could not break more than two rounds. Due to the high trail weight along with the reasonably strong Σ function we reason that the DMWT hash provides sufficient strength to be used as a cryptographic one-way hash function.

6 Conclusion

We have presented a new data transformation CSQUARE, along with a strong argument concerning the proof of the minimal multi-round trail weight. We have shown that for several dimensions the CSQUARE design yields a higher four round trail weight than the SQUARE design. We have also argued that the CSQUARE design posesses a fast and spacewise efficient quality when implemented in hardware.

As a demonstration of the design in use we have presented the DMWT 512-bit cryptographic one-way hash function. Based on the minimal trail weight of the CSQUARE design we have proven that DMWT emits no differential or linear bias with a probability above uniformly random.

Future work should concentrate on proving the general case of k for the second theorem as well as seeking a more efficient method of implementation. I would like to thank Jean-Luc Cooke for some helpful comments during the initial research period of this project.

References

[1] J. Daemen and L. Knudsen and V. Rijmen, "The Block Cipher SQUARE", Fast Software Encryption, v.1267 of Lecture Notes in Computer Science, pp. 149–165. Springer-Verlag, 1997

[2] Paulo S.L.M. Barreto and Vincent Rijmen, "The WHIRLPOOL Hashing Function"

[3] J. Stern and S. Vaudenay. CS-Cipher. In Fifth International Workshop on Fast Software Encryption, Berlin, Germany, March 1998. Springer-Verlag.

[4] S.Vaudenay, "On the Security of the CS-Cipher", Fast Software Encryption, March 1999, Springer-Verlag, pp. 260-274

[5] B. Preneel, R. Govaerts and J. Vandewalle, "Hash Functions based on block ciphers: a synthetic approach.", Crypto '93, Springer-Verlag, 1993, pp. 368–378

[6] S. Vaudenay, "On the Need for Multipermutations: Cryptanalysis of MD4 ad SAFER", LIENS - 94 - 23, November 1994

[7] Tom St Denis, "Fast Pseudo-Hadamard Transforms", Cryptology ePrint Archive, Report 2004-010

Appendix A - Test Vectors

The following are test vectors for our DMWT hash function.

```
""
        e2 26 96 5f 94 47 a1 38
        21 9a 04 6e 2f 9b 45 33
        4e c2 a8 f2 1b 33 db 98
        b6 5c cc 04 f7 55 24 0c
        1c 65 1f 7f 15 c1 5b 23
        00 16 aa 00 21 4c bc ef
        68 56 fb cb e5 48 84 53
        66 38 92 56 e4 c0 62 d3

"abc"
        ae 25 e1 33 c5 61 d7 30
        1d 35 0e 67 15 54 72 f5
        77 1c a2 91 b6 39 f0 53
        b1 67 fd 93 75 ae 4c 94
        66 96 97 ac 9b 2f de 6b
        b9 87 dd a7 1d f4 ef ff
        4b 32 b3 4e ba 69 ae 75
        30 d9 74 f9 e7 64 9c 43

"aaaabbbb" x 12 times
        43 1d c3 9a 4b a5 47 6e
        de 6f 19 bb 7e 6e 97 57
        60 d5 d2 c0 a8 32 7c 63
        38 a7 75 d4 07 b2 78 a0
        a4 90 a0 bb 07 53 00 21
        60 e8 27 9f 83 3c fe 34
        4c 1a 0b 2e 9d 39 db 34
        8a aa 0f 65 d0 d4 0c 62
```