# Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 8 (21): 3863-3870, 2008
ISSN 1812-5654


# A New Message Authentication Technique Using Zigzag Manipulation and Block Chaining

Mohammad Abbadi
Department of Information Technology, Mut'ah University, Jordan

**Abstract:** This study introduces a new implementation based on DES block cipher technique and the major three functions used in MD5, The idea is based on taking some outputs from every DES loop, applying functions over them, then, finally applying chaining functions on the ciphered message blocks. This study is aimed to introducing a new message authentication technique that takes the output of DES Iterations as an input, manipulates it through a zigzag and some other diffusion operations to produce the basic of a MAC code. The proposed technique encompasses an efficient performance and inherits the DES strength points. Moreover, the technique surmounts most of the CBC MAC chaining weaknesses; this is done by changing the $K_1$ and $K_2$ values for every iteration.

**Key words:** DES block cipher, message authentication code, cipher application, EAX, CCM, DES, zigzag

## INTRODUCTION

Encryption is used to provide privacy of data, that is, to keep the data secret from people other than the intended recipients (Jutla, 2001). It has a significant advantage, which is the encryption operations don't depend on each other there's no chaining (Rogaway and Shrimpton, 2006). Message authentication, also often called message integrity, provides assurance to the recipient of the data that it came from the expected sender and has not been altered in transit (Black and Rogaway, 2005).

The reality is that you need both mechanisms, even to get just privacy. Using both encryption and message integrity and they're both somehow based on cryptographic primitives, obviously you should be able to make savings by somehow combining common operations. It turns out that you can, but it's a much more subtle process than you might at first think messages (Dworkin, 2005; Rogaway and Shrimpton, 2006; Boneh *et al*., 2004; National Institute of Standards and Technology, 2001; Lee, 1999).

Encryption alone isn't sufficient to ensure privacy. Several researches provide a number of failure modes in IPsec when encryption alone is applied (Ferguson, 2002). The 802.11 WEP (Wired Equivalent Privacy) demonstrates this failure in a very simple manner (Cam-Winget *et al*., 2003; IEEE, 1999). Researchers had a short discussion with links about other reasons for always using a message authentication code, not only when message integrity is required, but also when data confidentiality is needed were instigated (Rogaway and Shrimpton, 2006; Bellare *et al*., 1996; National Institute of Standards and Technology, 1985).

**Message authentication code:** To guarantee message integrity, you can either use public-key digital signatures (which are computationally very expensive and have other problems) or you can use a MAC (Message Authentication Code). A MAC can be thought of as a keyed hash. There are a number of different ways to create MACs such as HMAC, CMC-MAC and Carter-Wegman MAC (Black and Rogaway, 2005; Dworkin, 2005; Cary and Venkatesan, 2003; IEEE, 1999).

In combined modes using block ciphers, two lots of processing are used on the data, one to encrypt it and one compute a MAC. While thinking about all of this, two new requirements arose:

- It should be possible to MAC a whole packet, while only encrypting part of it.
- It should be possible to parallelize the underlying operations, to enable high throughput if you're prepared to throw extra hardware at the problem.

The current IPsec solutions, based on CBC mode encryption and HMAC, take care of the first of these, but not the second; both operations are inherently sequential in nature because of the chaining (Black and Rogaway, 2005). After the multiple security problems with 802.11 WEP, there was pressure on the IEEE standards committee to do something secure for a change (Ferguson, 2002; IEEE, 1999). This committee didn't have the kind of cryptographic expertise needed to design the solution from the ground up, as they had already demonstrated, so, they wanted to adopt a single, easy-to-use package solution. OCB mode was proposed but patent considerations apparently prevented its adoption (Rogaway *et al*., 2001).

**CCM:** CCM (Counter with CBC-MAC) mode was eventually specified by a group of consultants and adopted for the next generation of 802.11 wireless security. (There's also an intermediate patch that many manufacturers have adopted, but I'm referring here to the real solution.) CCM is, cryptographically speaking, quite straightforward (Rogaway and Wagner, 2003). The supplied key is used to derive separate keys for encryption and integrity protection, then the encryption is done using counter mode, while the MAC is calculated using CBC-MAC, both as mentioned above. A header is prep ended to the data, including the length of the message and associated unencrypted data. Because this mode was specifically designed for 802.11, it meets those requirements very well, but has been criticized for general purpose use on the grounds that the header means you can't process a stream of data (you need to know the length of it before you can begin processing), it can't be parallelized (because of the use of CBC-MAC) and the format of the header is a bit fiddly (to keep it very small based on 802.11 requirements) (Dworkin, 2005).

**EAX:** EAX (Encrypt then Authenticate then Translate) mode is very similar in concept to CCM mode and was developed to address the criticisms mentioned earlier. These authors coined a new term, AEAD (Authenticated Encryption with Associated Data) for the problem of encrypting partial packets while authenticating all of them (Bellare *et al*., 2004). In essence, this method still boils down to using Counter Mode to encrypt the data and CBC-MAC to create the MAC. The last block of input for the MAC is processed in a special manner to prevent attacks that work by appending data, but without introducing a dependency on the length of the data or extending it with unnecessary padding. Again, because of the use of CBC-MAC, it still cannot be completely parallelized (Black and Rogaway, 2005; Ferguson *et al*., 2003; Bellare *et al*., 1996).

## THE PROPOSED TECHNIQUE (ZBM)

**The key:** THE ZBM fundamentally uses a 256 bits key which will be used also to extract 2 Numbers $K_1$ and $K_2$. $K_1$'s value ranges between 1 and 256 and $K_2$'s value ranges between 1 and 16.

However, you may just set the 2 values manually, this will also produce $K_1$ and $K_2$.

**The ZBM description:** The technique that will be used to produce the ZBM based on DES key is as follow:

First, if $K_1$ and $K_2$ are not detected calculate them as shown in Fig. 1.

Second, the first block will be zigzagged in the file based on $K_1$ value, so that $K_1$-1 cells will be skipped in the zigzag then starts zigzagging from the $K_1$'th Cell up to the end of the block then go back to the skipped $K_1$-1 values as shown in Fig. 2 where $K_1$ = 25 and $K_2$ = 6, this will distribute the original key bytes in fair way and will be useful in the next step which is to produce the 2k numbers $K_1$ and $K_2$.

Figure 3 shows how good the zigzag operation as a technique of distribution is. The image shows the Permutation in case of starting from the first position but when starting from another position it will be totally different one.

Third, regenerate the 2 k numbers $K_1$ and $K_2$ by apply the first step on the generated block. These keys will be used for the next block as shown in Fig. 4.

Fourth, the generation of the 263 bit ZBM key is based on the DES key. This process stands on counting

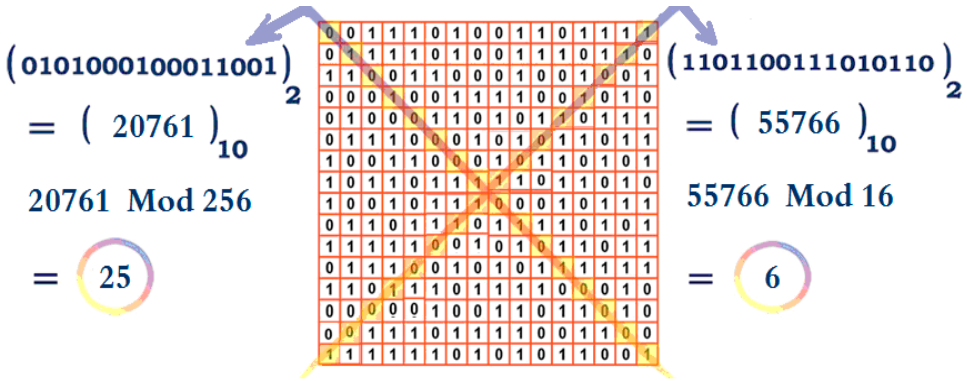


Fig. 1: Calculating $K_1$ and $K_2$ values

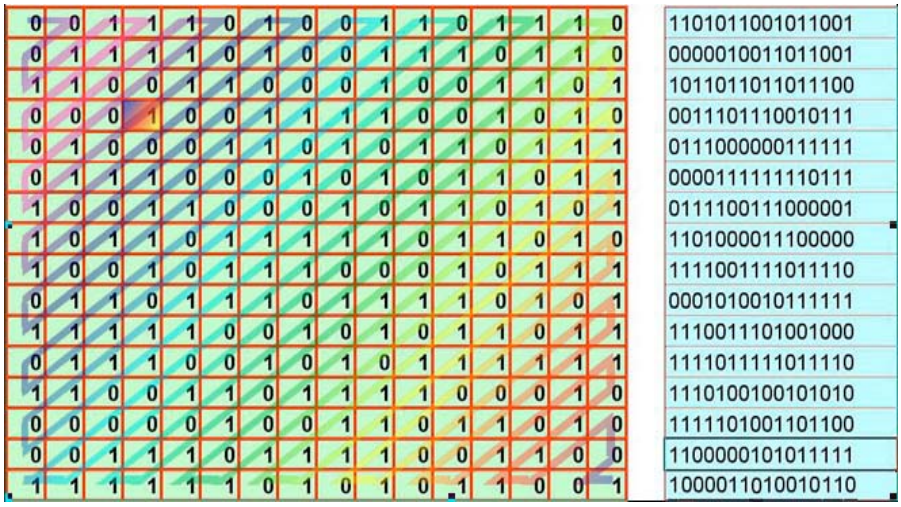


Fig. 2: Starting the zigzag process from location $K_1$ (25) of the zigzag

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
| 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 |
| 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 |
| 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 | 96 |
| 97 | 98 | 99 | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 | 112 |
| 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 | 128 |
| 129 | 130 | 131 | 132 | 133 | 134 | 135 | 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 | 144 |
| 145 | 146 | 147 | 148 | 149 | 150 | 151 | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 | 160 |
| 161 | 162 | 163 | 164 | 165 | 166 | 167 | 168 | 169 | 170 | 171 | 172 | 173 | 174 | 175 | 176 |
| 177 | 178 | 179 | 180 | 181 | 182 | 183 | 184 | 185 | 186 | 187 | 188 | 189 | 190 | 191 | 192 |
| 193 | 194 | 195 | 196 | 197 | 198 | 199 | 200 | 201 | 202 | 203 | 204 | 205 | 206 | 207 | 208 |
| 209 | 210 | 211 | 212 | 213 | 214 | 215 | 216 | 217 | 218 | 219 | 220 | 221 | 222 | 223 | 224 |
| 225 | 226 | 227 | 228 | 229 | 230 | 231 | 232 | 233 | 234 | 235 | 236 | 237 | 238 | 239 | 240 |
| 241 | 242 | 243 | 244 | 245 | 246 | 247 | 248 | 249 | 250 | 251 | 252 | 253 | 254 | 255 | 256 |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 17 | 33 | 18 | 3 | 4 | 19 | 34 | 49 | 65 | 50 | 35 | 20 | 5 | 6 |
| 21 | 36 | 51 | 66 | 81 | 97 | 82 | 67 | 52 | 37 | 22 | 7 | 8 | 23 | 38 | 53 |
| 68 | 83 | 98 | 113 | 129 | 114 | 99 | 84 | 69 | 54 | 39 | 24 | 9 | 10 | 25 | 40 |
| 55 | 70 | 85 | 100 | 115 | 130 | 145 | 161 | 146 | 131 | 116 | 101 | 86 | 71 | 56 | 41 |
| 26 | 11 | 12 | 27 | 42 | 57 | 72 | 87 | 102 | 117 | 132 | 147 | 162 | 177 | 193 | 178 |
| 163 | 148 | 133 | 118 | 103 | 88 | 73 | 58 | 43 | 28 | 18 | 14 | 29 | 44 | 59 | 74 |
| 89 | 104 | 119 | 134 | 149 | 164 | 179 | 194 | 209 | 225 | 210 | 195 | 180 | 165 | 150 | 135 |
| 120 | 105 | 90 | 75 | 60 | 45 | 30 | 15 | 16 | 31 | 46 | 61 | 76 | 91 | 106 | 121 |
| 136 | 151 | 166 | 181 | 196 | 211 | 226 | 241 | 242 | 227 | 212 | 197 | 182 | 167 | 152 | 137 |
| 122 | 107 | 92 | 77 | 62 | 47 | 32 | 48 | 63 | 78 | 93 | 108 | 123 | 138 | 153 | 168 |
| 183 | 198 | 213 | 228 | 243 | 244 | 229 | 214 | 199 | 184 | 169 | 154 | 139 | 124 | 109 | 94 |
| 79 | 64 | 80 | 95 | 110 | 125 | 140 | 155 | 170 | 185 | 200 | 215 | 230 | 245 | 246 | 231 |
| 216 | 201 | 186 | 171 | 156 | 141 | 126 | 111 | 96 | 112 | 127 | 142 | 157 | 172 | 187 | 202 |
| 217 | 232 | 247 | 248 | 233 | 218 | 203 | 188 | 173 | 158 | 143 | 128 | 144 | 156 | 174 | 189 |
| 204 | 219 | 234 | 249 | 250 | 235 | 220 | 205 | 190 | 175 | 160 | 176 | 191 | 206 | 221 | 236 |
| 251 | 252 | 237 | 222 | 207 | 192 | 208 | 223 | 238 | 253 | 254 | 239 | 224 | 240 | 255 | 256 |

Fig. 3: How zigzag distributes bits incase it starts from the first location

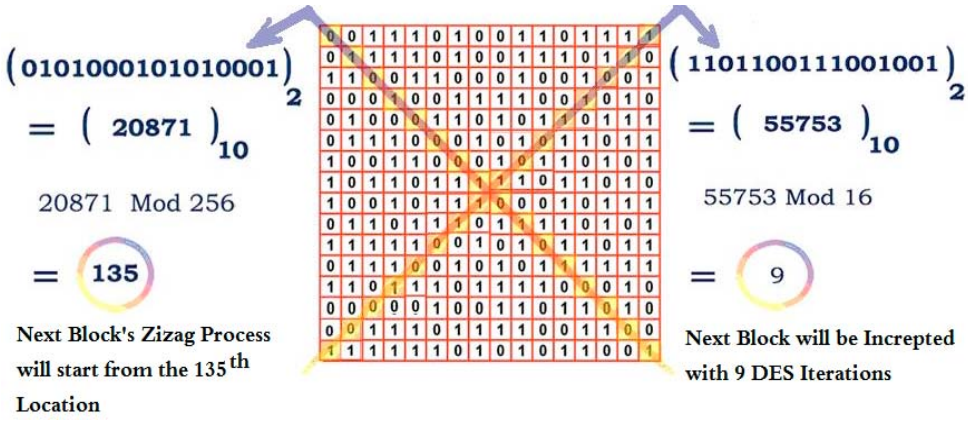


Fig. 4: Calculating $K_1$ and $K_2$ values which will be used for the next block

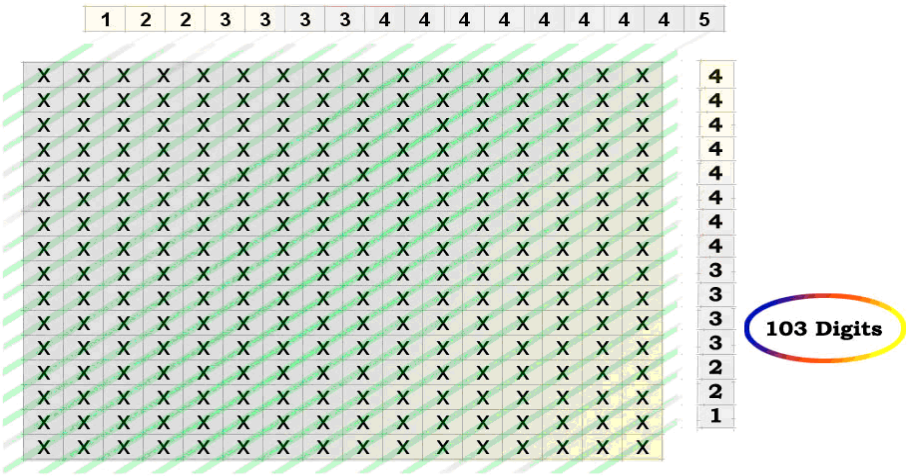


Fig. 5: Counting the number of one-bits in the diagonal order

the number of one-bits in the diagonals, in the rows and finally in the columns as shown in Fig. 5-7, respectively. Note that the same process will be used to produce the ZBM block in every iteration. These values will be saved as binary by dedicating proper sizes of bits for all entries. The number of bits dedicated for each entry was chosen based on the max value for that entry. Figure 8 shows how every cell is represented within three entries and Fig. 9 shows how the 263 bits key is collected.

Fifth, using the above 263 bits key and the 2 numbers $K_1 = 25$ and $K_2 = 6$, $K_1$ will force zigzagging process of the First ZBM block to start from location 25 and $K_2$ will guide the algorithm to take the next block data from the 6th DES iteration. However, these two values will be changed for the next block and for every block after it as shown in the third step, this will give the algorithm its strength, so that if 1 bit was modified in any block, then this will give a fully different ZBM block values.

Sixth, once the 263 bits block is produced, the chaining process is started using addition and mod operations on parts of 25 bits of the block as shown in Fig. 10.

Chaining in the ZBM is not like the normal CBC MAC chaining, as new factors effect on the blocks chaining process, Fig. 11 shows how can this be achieved.

**Advantages of the ZBM:** The ZBM inherits DES strength points, this include all the security features of block cipher in general, it also add to this the strength of its longer key of 263 bits and the two K-values which will

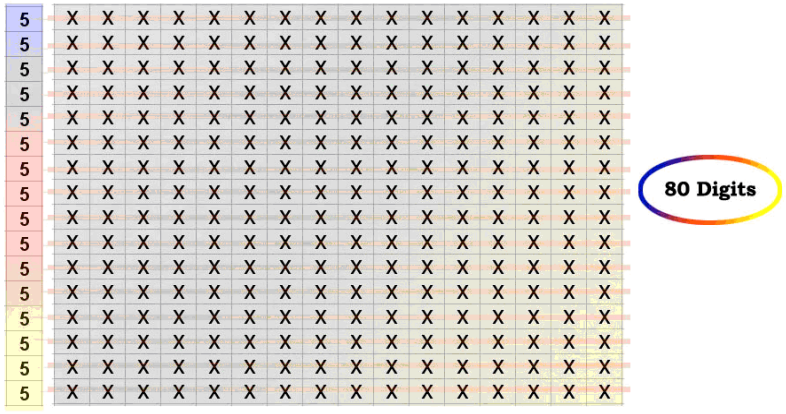


Fig. 6: Counting the number of one-bits in the row order

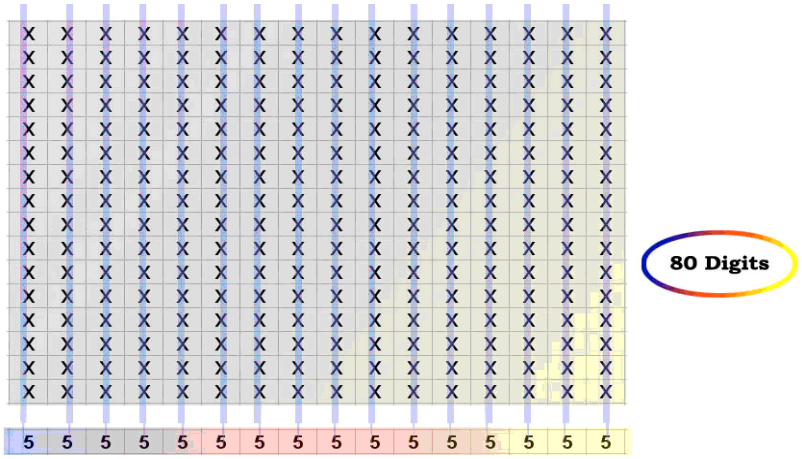


Fig. 7: Counting the number of one-bits in the column order

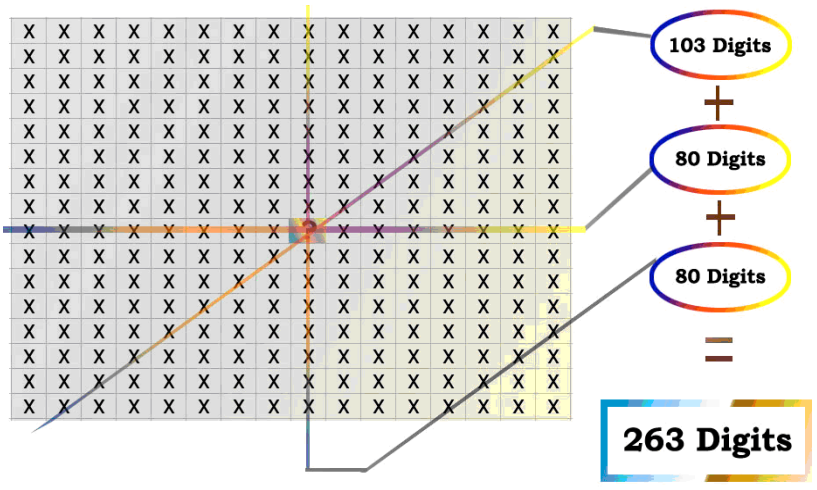


Fig. 8: How every cell is represented within three entries

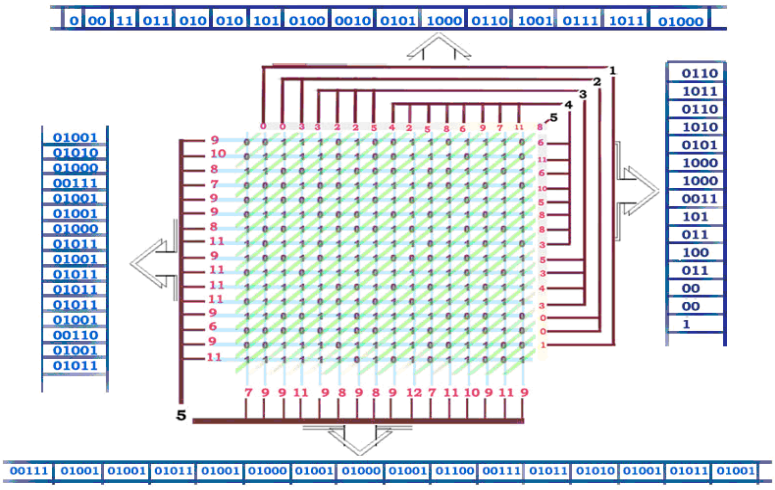

Fig. 9: How the 263 bits key is collected

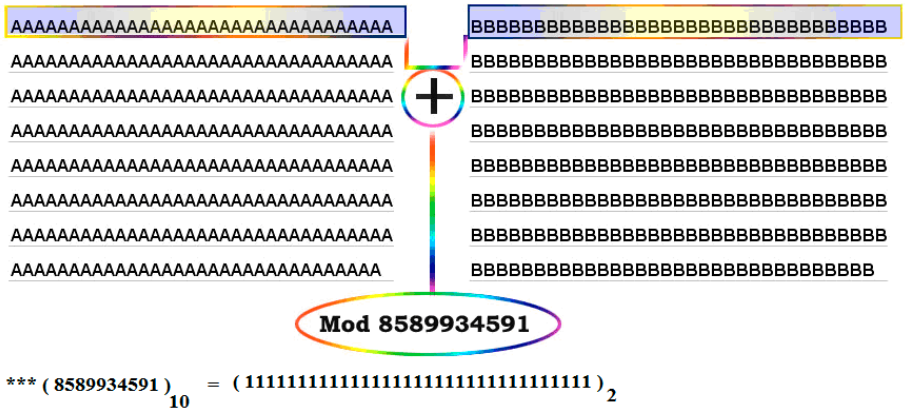


Fig. 10: The chaining process using addition and mod operations on parts of 25 bits of the block

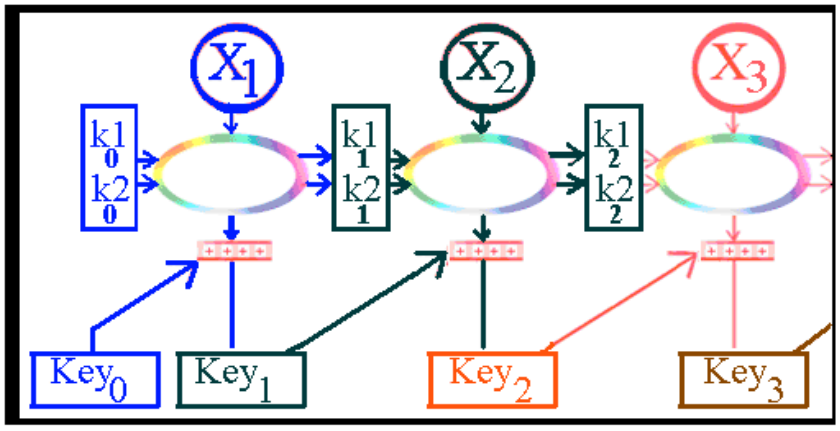


Fig. 11: Calculating $K_1$ and $K_2$ values which will be used for the next block

absolutely make it much more difficult to inference the original message based on the final ZBM.

By choosing just one DES iteration for each block in the message with a strong function which is difficult to revert, more strength will be added and also MACing time will be reduced. Note that being based on DES algorithm here does not mean that the encryption has to be applied when the message is MACed, just one iteration of DES is used for each block of the message in a semi random but strict technique. This technique can surmount most of the CBC MAC chaining drawbacks; this is done by using different values of $K_1$ and $K_2$ which make dropping or changing one bit from the message a reason for changing the whole ZBM value.

**Drawbacks of the ZBM:** Still the values of $K_1$ and $K_2$ between 0 and 256 is a very small number and make the ZBM subject to brute forcing the block size on which the ZBM and DES which ZBM is based on-is a small one However, this do not mean that the technique as a whole is a weak one. Comparing to the well known MACing and hash function the 256 bits block size is a small size, but this can be counted as an advantage of the technique.

## CONCLUSION AND FUTURE WORK

Cryptography is hard to get right and it has recently become clear that combining encryption and authentication to get practical security is downright difficult. This study introduced a new method that can be ensuring a minimum of Authenticity and work effectively. ZBM initiated some modifications on the Chaining MAC with DES algorithm; this gave a light on how stronger cipher algorithms like AES, RSA and blowfish can be exploited with some strict process toward stronger MAC algorithms.

## REFERENCES


Bellare, M., R. Canetti and H. Krawczyk, 1996. Keying hash functions for message authentication. Advances in Cryptology-Proceedings of Crypto 1996, LNCS 1109, January 01, Springer-Verlag, Berlin, Germany, pp: 1-15.

Bellare, M., P. Rogaway and D. Wagner, 2004. The EAX mode of operation. Proceedings of the 11th Workshop on Fast Software Encryption (FSE 2004), LNCS 3017, 18th Jan., Springer-Verlag, Berlin, Germany, pp: 389-407.

Black, J. and P. Rogaway, 2005. CBC MACs for arbitrary-length messages: The three-key constructions. J. Cryptol., 18: 111-131.

Boneh, D., G. Di Crescenzo, R. Ostrovsky and G. Persiano, 2004. Public key encryption with keyword search. Advances in Cryptology-Proceedings of Eurocrypt 2004, LNCS 3027, May 2-6, Springer-Verlag, Berlin, Germany, pp: 506-522.

Cam-Winget, N., R. Housley, D. Wagner and J. Walker, 2003. Security flaws in 802.11 data link protocols. Commun. ACM., 46: 35-39.

Cary, M. and R. Venkatesan, 2003. A message authentication code based on unimodular matrix groups. Advances in Cryptology-Proceedings of Crypto 2003, LNCS 2729, October 17, Springer-Verlag, Berlin, Germany, pp: 500-512.

Dworkin, M., 2005. Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication. National Institute of Standards and Technology, Special Publication 800-38B. csrc.nist.gov/publications/nistpubs/800-38C/SP800-38B.pdf.

Ferguson, M., 2002. An improved MIC for 802.11 WEP. IEEE doc. 802.11-02/020r0, Jan. 17, http://www.grouper.ieee.org/groups/802/11/Documents/DocumentHolder/2-020.zip.

Ferguson, M., D. Whiting, B. Schneier, J. Kelsey and T. Kohno, 2003. Helix fast encryption and authentication in a single cryptographic primitive. Pre-Proceedings of the 10th International Workshop on Fast Software Encryption (FSE2003), LNCS 2887, February 2003, Springer-Verlag, Berlin, Germany, pp: 345-362.

IEEE, 1999. ISO/IEC 8802-11 ANSI/IEEE Std 802.11-1999. Information technology telecommunications and information systems-local and metropolitan area networks-specific requirements-Part 11: Wireless LAN Media Access Control (MAC) and Physical Layer (PHY) Specifications.

Jutla, C., 2001. Encryption modes with almost free message integrity. Advances in Cryptology-Proceedings of Eurocrypt 2001, LNCS 2045, January 01, Springer-Verlag, Berlin, Germany, pp: 529-544.

Lee, A., 1999. Guideline for Implementing Cryptography in the Federal Government. National Institute of Standards and Technology, Special Publication 800-21. http://csrc.nist.gov/publications/nistpubs/index.html.

National Institute of Standards and Technology (NIST), 1985. FIPS Pub 113: Computer Data Authentication. http://www.itl.nist.gov/fipspubs/fip113.htm.

National Institute of Standards and Technology (NIST), 2001. FIPS Pub 197: Advanced Encryption Standard (AES). csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

Rogaway, P., M. Bellare and J. Black, 2001. OCB: A block-cipher mode of operation for efficient authenticated encryption. ACM Trans. Inform. Syst. Security, 6: 365-403.

Rogaway, P. and D. Wagner, 2003. A critique of CCM. J. Cryptol., 13: 315-338.

Rogaway, P. and T. Shrimpton, 2006. Deterministic authenticated-encryption: A provable-security treatment of the keywrap problem. Advances in Cryptology-Proceedings of Eurocrypt 2006, LNCS 4004, July 04, Springer-Verlag, Berlin, Germany, pp: 315-338.