Defence Science Journal, Vol. 66, No. 6, November 2016, pp. 582-589, DOI : 10.14429/dsj.66.10798

Cryptanalytic Attacks on International Data Encryption Algorithm Block Cipher

Harish Kumar Sahu[*,!],Vikas Jadhav[#], Shefali Sonavane[#], and R.K. Sharma[!]
[!]*Department of Mathematics, Indian Institute of Technology Delhi, New Delhi - 110 016, India*
[#]*Department of Information Technology, Walchand College of Engineering, Sangli - 416 415, India*
[*]*E-mail: harish.sahu@gmail.com*

ABSTRACT

International data encryption algorithm (IDEA) is a secret key or symmetric key block cipher. The purpose of IDEA was to replace data encryption standard (DES) cipher, which became practically insecure due to its small key size of 56 bits and increase in computational power of systems. IDEA cipher mainly to provides data confidentiality in variety of applications such as commercial and financial application e.g. pretty good privacy (PGP) protocol. Till 2015, no successful linear or algebraic weaknesses of IDEA of have been reported. In this paper, author explained IDEA cipher, its application in PGP and did a systematic survey of various attacks attempted on IDEA cipher. The best cryptanalysis result which applied to all keys could break IDEA up to 6 rounds out of 8.5 rounds of the full IDEA cipher[1]. But the attack requires 2^{64} known plaintexts and $2^{126.8}$ operations for reduced round version. This attack is practically not feasible due to above mentioned mammoth data and time requirements. So IDEA cipher is still completely secure for practical usage. PGP v2.0 uses IDEA cipher in place of BassOmatic which was found to be insecure for providing data confidentiality.

Keywords: International data encryption algorithm, secret key, symmetric key block cipher, cryptanalysis

1. INTRODUCTION

International data encryption algorithm (IDEA) is 128 bit key size symmetric block cipher. It was designed by Xuejia and James of ETH-Zürich which operates on 64-bit plaintext data block[2]. The purpose of IDEA was to replace data encryption standard (DES) cipher, which became practically insecure due to its small key size of 56 bits and increased computational power. It is included in pretty good privacy (PGP)[3] to provide data security. It was designed in the beginning of the 90's by mixing three different algebraically incompatible operations. Using these three operations a strong round function is constructed and this round function operates eight times on plaintext. IDEA is designed to provide more of security as compared to speed.

2. THE IDEA BLOCK CIPHER

IDEA is a symmetric key block cipher which uses 64 bits block of data at a time for encryption using 128 bit key. It was intended to replace DES cipher. So, design of IDEA was motivated with the goal of removing the issues with DES cipher. DES uses small key size of 56 bits. So IDEA was designed with 128 bits of key which made it quite difficult to exhaustively try all possible keys in far future. Diffusion is a desirable cryptographic property which is basically means that each plaintext bit should affect all bits of cipher text and also each key bit should affect all the cipher bits. Diffusion hides the statistical properties of plaintext and key in ciphertext. IDEA exhibits very good diffusion property using multiplication and addition (MA) structure (in Fig. 1) which has two additions and two multiplication operations in it. The MA structure is applied in each round of the cipher. In Confusion, the ciphertext is dependent on plaintext and key is in very complicated and involved way. Confusion was achieved in DES using XOR and small nonlinear 6x4 substitution boxes while IDEA uses three different unrelated operations.

Overall IDEA exhibits good cryptographic properties, ease of implementation and resists all existing cryptanalysis techniques. The IDEA block cipher consists of 8 identical rounds in which 64-bit block of data and 128-bit key is used (Fig. 2. illustrates one round). One single round uses of 6 sub-keys and each sub-key consist of 16 bit word (totaling 6*16=96 bits). These six sub-keys for a particular k-th round is denoted by Z_j^k, where $0 \leq j \leq 5$. After such 8 rounds, output transformation is performed which is often called half round. The k-th round transforms a 64-bit input vector $(X_0^k, X_1^k, X_2^k, X_3^k)$ to an output vector $(W_0^k, W_1^k, W_2^k, W_3^k)$ where each vector element is 16 bit word. Rounds keys are derived from the 128-bit main master key by shifting it circularly left, which

Received : 22 February 2016, Revised : 01 July 2016
Accepted : 16 August 2016, Online published : 31 October 2016

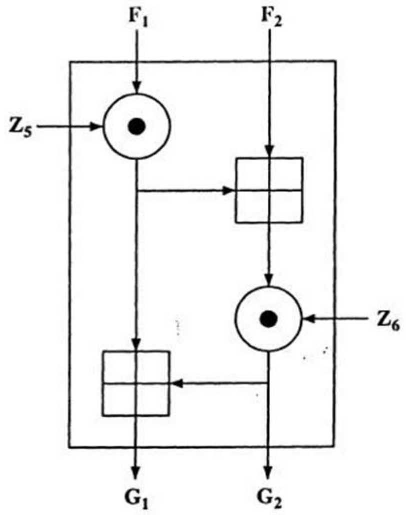

Figure 1. Multiplication addition structure (MA).

will be explained in detail in next section of key scheduling. In each round of operations mixing these three algebraically incompatible group operations are performed.

- Bit-by-bit Exclusive OR denoted as ⊕ .
- Addition of unsigned integers modulo 2^{16} (65536), denoted as ⊞ .
- Multiplication of unsigned integers modulo $2^{16}+1$ (65537) denoted as ⊙ .

Round *k* is defined by following operations:

First of all calculate two intermediate values:

$$a^k = \left(X_0^k \odot Z_0^k\right) \oplus (X_2^k \boxplus Z_2^k),$$

$$b^k = (X_1^k \boxplus Z_1^k) \oplus (X_3^k)$$

These two values are input of multiplication-addition (MA) box which results in:

$$c^k = (((a^k \odot Z_4^k \boxplus) b^k) \odot Z_5^k)$$

$$d^k = (\left(a^k \odot Z_4^k\right) \boxplus a^k)$$

The output of k-th round is obtained through:

$$W_0^k = ((X_0^k \odot Z_0^k) \oplus c^k),$$

$$W_1^k = ((X_2^k \odot Z_2^k) \oplus c^k),$$

$$W_2^k = ((X_1^k \odot Z_1^k) \oplus d^k),$$

$$W_3^k = ((X_3^k \odot Z_3^k) \oplus d^k)$$

After the 8 rounds final key whitening layer is applied:

$$W_0^9 = \left(X_0^9 \odot Z_0^9\right);$$

$$W_1^9 = (X_2^9 \boxplus Z_1^9);$$

$$W_2^9 = (X_1^9 \boxplus Z_2^9);$$

$$W_3^9 = (X_3^9 \odot Z_3^9)$$

2.1 Sub-Key Schedule

Each round of IDEA requires 6 sub-keys and there are 8.5 rounds, so in total 8*6=48 keys of eight rounds and 4 keys of output transformation i.e. 52 (48+4) keys are needed. Main key of IDEA is of 128-bit, which is divided into 8 sub-keys 16-bits each. First round extracts 6 sub-keys from these 8 keys. Followed by circular left shift of the key by 25 bits, then next eight sub-keys are extracted in the similar fashion. This process is repeated until all 52 required sub-keys are generated as shown in Table 1. Here, Z_1^k indicates first sub-key where *k* is round number and 1 shows key number of round. Table 1, shows how 128-bit key is scheduled with respect to each round with 25 bit circular left shift.

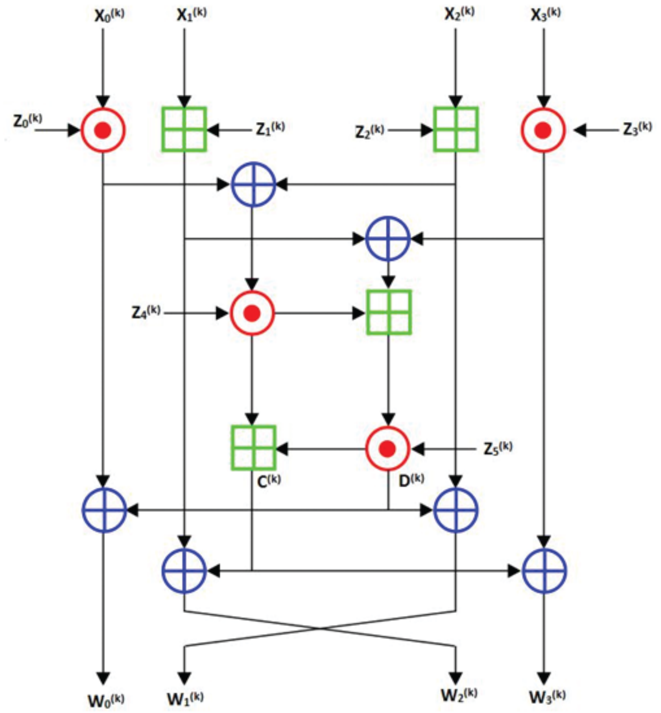

Figure 2. Round k. of IDEA.

2.2 Application of IDEA in PGP

Zimmermann[3] alone designed and developed pretty good privacy (PGP) in 1991. PGP provides mainly confidentiality, integrity, compression and authentication for electronic mail (Email) and file storage. PGP is now an RFC 3156. It uses four different kinds of keys i.e. one time session keys, public keys, private keys and passphrase based symmetric keys.

Table 1. Key-Schedule of IDEA Cipher. K[0-15] denotes bits from 0 to 15 of 128-bit key (K)

	Z_1^k	Z_2^k	Z_3^k	Z_4^k	Z_5^k	Z_6^k
1	K[0-15]	K[16-31]	K[32-47]	K[48-63]	K[64-79]	K[80-95]
2	K[96-111]	K[112-127]	K[25-40]	K[41-56]	K[57-72]	K[73-88]
3	K[89-97]	K[105-120]	K[121-8]	K[9-24]	K[50-65]	K[66-81]
4	K[82-97]	K[98-113]	K[114-1]	K[2-17]	K[18-33]	K[34-49]
5	K[75-90]	K[91-106]	K[107-122]	K[123-10]	K[11-26]	K[27-42]
6	K[43-58]	K[59-74]	K[100-115]	K[116-3]	K[4-19]	K[20-35]
7	K[36-51]	K[52-67]	K[68-83]	K[84-99]	K[125-12]	K[13-28]
8	K[29-44]	K[45-60]	K[61-76]	K[77-92]	K[93-108]	K[109-124]
8.5	K[22-37]	K[38-53]	K[54-69]	K[70-85]	-	-

One time session keys are used as an encryption key for the block ciphers used in PGP. Public and private keys are mainly used for providing authentication plus to securely transfer the session keys between communicating parties. Integrity of data is basically to ensure that data has not been modified in transit accidentally or maliciously, and is achieved by using hash function SHA-1 in PGP. It uses symmetric key encryption because it is faster as compared to asymmetric key encryption. But uses asymmetric key cryptography for exchange of small session keys which are used as keys for symmetric block cipher.

Confidentiality: Confidentiality is achieved by encrypting the information. In case of PGP three symmetric block ciphers are available for encryption e.g. CAST-128, Triple DES and IDEA. IDEA is widely used option for encryption.

Below the steps are described how IDEA cipher is used for providing confidentiality in PGP.

(a) It generates a random string of 128 bits which will be used as a key for IDEA cipher. For every session a different key is generated.
(b) Data (email or file) is encrypted using IDEA cipher in Cipher Feedback (CFB) mode. In CFB mode only 64 bits of cipher are fed. Data is encrypted in block sizes of 64 bits at a time.
(c) Key is encrypted using RSA or DSS public key of intended receiver and sent it along with encrypted data to receiver.
(d) Receiver first decrypts the session key by applying his RSA or DSS private key.
(e) Finally session key is used to decrypt the data using IDEA decryption.

PGP success is mainly because it is an open standard. PGP runs on most of the platforms flawlessly such as Windows, Linux or Mac. It uses best in class cryptographic algorithms for providing the authentication and confidentiality. These algorithms used in PGP are widely tested by public over a long period of time and considered secure.

PGP provides authentication using RSA / DSS / Diffie Hellman key exchange, compression of data using zip, integrity of data using SHA-1 hash function. PGP also provides segmentation of data when email data is more than 50 K octets.

PGP achieves confidentiality of data using IDEA cipher. Since IDEA cipher has key size of 128 bits which provide security of order 2^{128}. To try all 2^{128} that is 340 282 366 930 938 463 463 374 607 431 768 211 456 different keys systematically will take 1000 of years for even the fastest super computer of the world.

2.3 Important attacks on Block Cipher

Ciphertext only: In cryptography, a ciphertext-only attack (COA) is the most difficult attack scenario from cryptanalytic point of view in which cryptanalyst is assumed to have only a few ciphertexts data with him. COA is the most difficult to cryptanalyse since no other side channel information is available. In case of practical COA, attacker still has few knowledge of the plaintext. Just like the attacker might know the language in which the plaintext is presented or the required distribution of strings or characters in the plaintext. Mostly standard protocol data or messages are part of the plaintext in many deployed systems and these can be easily guessed or known efficiently as part of COA on these systems. COA is considered successful if cryptanalyst manages to find corresponding plaintext data and if the key is found than it is a breakthrough. This ability to obtain any amount of information from the underlying ciphertext is considered as an achievement of attack.

Known Plaintext: The known-plaintext attack (KPA) is an attack scenario in which cryptanalyst has access to one or more plaintext-ciphertext pairs formed with the secret key which is unknown. Example of such scenario is mobile communication or network traffic in which it is known that first few packets or data is fixed and whatever initial encrypted data we are getting is corresponding ciphertext. The goal is to find the key or decrypt the rest of data for which plaintext is not known.

Chosen Plaintext: Chosen plaintext attack (CPA) is an attack scenario in which cryptanalyst has access to encryption box or oracle. Wherein cryptanalyst can choose any number of different plaintext messages and can get corresponding ciphertext (CT) message. So basically he does not the encryption key still he can encrypt the plaintext messages. In general, it does not seem a practical possible scenario. But nowadays ciphers are implemented in hardware and software, where we can get such access such as once we get the cipher device (hardware) or encryption box we can supply plaintext to it and get ciphertexts but key is not accessible since it is embedded inside the box. In case of public key cryptosystem where the encryption key is public, so we can encrypt any number of messages. Therefore certain attacks are possible using CPA in which user has multiple messages under same key and can retrieve the encryption key also.

Chosen Ciphertext: Chosen ciphertext attack (CCA) is an attack scenario in which the cryptanalyst can choose number of ciphertexts and supply it into decryption oracle/box and get corresponding plaintext but he does not know the actual decryption key used. CCA is a probable scenario since encryption and decryption functions of ciphers are implemented on hardware and software nowadays. Once attacker has the decryption box with him he can perform any number of decryption on the plaintexts of his choice. The goal here is not to figure out what is the message for given ciphertext but what key is being used in decryption function.

Chosen Text: In this attack, cryptanalyst uses combination of chosen plaintext and chosen ciphertext. Basically it is assumed that cryptanalyst has access to both the encryption and decryption oracle with him. Using this he can query oracle or black box for any given arbitrary plaintext and fetch corresponding ciphertext and vice versa. The goal of attacker is to just find the secret key, rest all information is available to him.

Differential Cryptanalysis: Is a technique to break certain types of ciphers. Biham and Shamir[4], described differential cryptanalysis, in 1990. They applied differential cryptanalysis on DES cipher. However, Coppersmith of TJ Watson Research Center claimed that the IBM researchers who designed DES were already aware about differential cryptanalysis. Differential cryptanalysis is more suitable in the chosen plaintext attack (CPA) scenario wherein attacker can choose the plaintext for encryption.The method searches for plaintext, ciphertext pairs whose difference is constant, in this method basically measure the difference between the two ciphertexts is measured as a function of the difference between the corresponding plaintexts. This difference is generally calculated by X-ORing the plaintexts pairs and X-ORing the ciphertexts pairs. E.g. the difference of two plaintexts M1 and M2 is defined as M1 $\oplus$ M2 bit-wise exclusive or (XOR) operation for DES. However the difference may be define in other ways too for other ciphers. Now the highest probability differential input is calculated, called characteristic which can be traced through several rounds of the cipher. Now assign these probabilities to the keys and locate the most probable key.

The Square Attack: Integral Cryptanalysis is also called SQARE attack because it was first designed and attempted by Khudsen[5] on block cipher SQARE, but the attack is not limited to the particular cipher SQARE. Knudsen showed it that the attack is more suitable to the block cipher based on SPN structures. Later on it was extended to the cipher having fiestal structure such as DES cipher and ARX based cipher also. The name integral cryptanalysis is taken from integral calculus. The X-OR difference of set of plaintext is kept 0 and by observing ciphertext difference he tried to figure out the operations of cipher in this attack.

Meet in the Middle Attack: Meet in the Middle (MITM) was first introduced by Diffie and Hellman. It is a generic attack applicable to several ciphers and does not depend on internals of the ciphers. It just requires the encryptions and decrypting oracle for getting the plaintext and its corresponding cipher text and vice versa. They demonstrated that double DES is prone to MITM. In double DES to increase security, normal DES encryption is applied two times with two different keys say K1, K2 of 56 bits each. It should provide security of $2^{(K1+K2)}$= 2^{56}+2^{56} = 2^{112}. However when MITM is applied on double DES its security is quite less as compare to 2^{112}.

Given a plaintext-ciphertext pair (P-C) of double DES, apply all 2^{56} choices of K1 on P with single DES encryption and store the result in sorted array. Then apply all choice of K2 on C with single DES decryption and search for each result in sorted array. Whenever we find a match, the particular value of K2 and its corresponding matched entry of K1 are the actual keys used in double DES. Total effort is just 2^{56} for forward encryption and 2^{56} for backward decryption. Search for matching is assumed to take constant time if applied with hash table for sorted result of K1 encryptions. Complexity is $2^{56} + 2^{56} = 2^{57}$ which is just two time of single DES, quite less than expected 2^{112}. They suggested using triple DES with two different keys, in order to achieve required security of 2^{112} which resist MITM also. MITM is suitably applicable because attacker can apply brute force on two DES encryption independently which it cannot do on triple DES.

Boomerang Attack: It was first introduced by Wagner in 1999, applied it to break COCONUT cipher designed in 1998. Using the boomerang attack, attackers are able to break the ciphers which were quite resistant to differential cryptanalysis. It differs from differential cryptanalysis in way that in order to cover complete cipher, it does not try to find a differential pattern having high probability but tries for several high probability patterns which may be un-related. The boomerang attack assumes access to both encryption and decryption oracles. Using the several high probable patterns it covers the whole cipher.

Biclique Attack: A biclique attack is a variant of the meet-in-the-middle (MITM) method of cryptanalysis which is applicable on both block cipher and hash functions. It was first proposed by Khovratovich, Rechberger and Savelieva and they applied it on AES-128 7 round version. Calculated complexity of order $2^{126.1}$ for full AES cipher. It utilises a biclique structure to increase the number of attacked rounds[6] in comparison to MITM.

3. ATTACKS ON IDEA CIPHER

Block ciphers play very crucial work to handle the confidentiality of data. Various cryptanalytic attacks have been attempted on IDEA since its inception. All the attacks presented so far deal with various reduced versions of IDEA. The best attack ever designed against IDEA is able to break 6 out of 8.5 rounds at $2^{126.8}$ computational cost which is practically slightly less than exhaustive search. When we consider reduced versions of IDEA, it is important to consider their size in increments of 0.5, depending on whether this component is used or not. The last round of output transformation is considered equivalent to 0.5 round.

3.1 Attack attempts on IDEA Block Cipher

3.1.1 Classical Attacks

Differential Cryptanalysis: This approach is generally applicable to all symmetric ciphers as well as to hash functions. In differential cryptanalysis we observe the differences in plaintext, how it passes through series of transformation in cipher and how it can affect the resultant difference at the ciphertext. Based on these we compute high probable differential, use to find the output of (r-1) round with certain probabilities, where r is total no. of rounds in cipher. Now cryptanalyst can find the last round key by exhaustively trying all possible keys for decryption of ciphertext and match it with output of (r-1) round computed earlier. In this attack pairs of plaintexts of certain constant differences are used. Then attacker computes the differences of cipher text in-order to detect some statistical patterns in their distribution. In 1993, Meier[7] used the above mentioned technique to cryptanalyse upto two and half round of IDEA which is quite faster than the brute force approach of trying all possible key choices.

Borst[8], *et al.* cryptanalyse IDEA using a differential linear attack (DLA). For three and half rounds they used truncated differentials. Nakahara et al. proposed another approach based on integral for two and half rounds version of IDEA.

3.1.2. The Square Attack

Integral cryptanalysis or SQAURE attack is also a chosen plaintext attack approach similar to differential cryptanalysis technique but integral cryptanalysis uses sets or multiple sets of plaintext ciphertext pairs X-OR difference whereas differential cryptanalysis use fixed plaintext ciphertext pairs X-OR difference. In integral or square approach sum of X-OR difference of input plaintext set is always zero, by observing the sum of ciphertext difference we try to figure out about the cipher operations. E.g. suppose input plaintext size is 64 bits we create set of plaintexts having their first 56 bits as a fixed pattern and for remaining last 8 bits we have all choices, which give a set of 256 such plaintexts. When we sum or X-OR these plaintext, result is always 0 for input difference. Now we calculate X-OR difference of corresponding ciphertexts which is non zero. Extract the information about cipher operations using these differences. Demirici[9] applied related key square attack on 2.5 round IDEA with time complexity of 2^{41}.

3.1.3. Simplified IDEA Variants

Some authors have attacked simplified version of IDEA, like Borisov[10], *et al.* in which they replaced all (Addition of unsigned integers modulo 216) operations with $\oplus$ (Bit-by-bit Exclusive OR) other than in last half round. They concluded that out of 2^{16} keys one key exists in a multiplicative differential characteristic over eight rounds which holds with probability of 2^{32}. Raddum[11] assumed light version of IDEA called IDEA-x/2, in which only 50 per cent operations per rounds got changed, while multiplication addition (MA) structure is not modified.

3.1.4. Meet In The Middle Attack

Demirici[6], *et al*. described meet in the middle[12] (MITM) attack on the reduced round version of IDEA cipher. For 4 and 4.5 round versions the attack provides a significant reduction in the attack's data complexity. The MITM attack on Block cipher considers a intermediate value V which can be computed from given plaintext and will be the only part of secrete key considering it to be Kt and it is easy to find from given cipher text and possibly will be different part of key as Kb[13]. The attack requires several pairs of known plaintext cipher text for guessing of Kt computed from plaintext and corresponding V values and store them in data structure called hash table which provide faster search. After that match of Kb is searched over hash table.

(a) Pre-computation is the first step of attack in which for a few chosen plaintexts it computes all choices of LSBs of the result C2 $\oplus$ C3, where C2 $\oplus$ C3 is ex-OR of second and third sub blocks of ciphertext. Then some plaintext ciphertext pairs are meticulously selected. These are partially decrypted using the different candidate keys. If the correct key is chosen for partial decryption than surely there is a match occurs in the set of pre computation and set of partial decryption process. It is very rare to get a match between these sets by a mere coincidence. This condition helps in discarding the incorrect candidate keys.

(b) CCA is also applicable for decipher function of IDEA. In order to determine more unknown decryption subkeys, there are two possible scenarios. First if reduced round cipher includes an extra half round in end than attack is applied exactly same as on normal cipher only the decryption keys are used in place of encryption keys. Second when reduced round version does not have that extra half round at end of cipher than attack need minor changes in multiplication addition half round. For three round IDEA, a set of 224 ciphertexts is generated using chosen ciphertext attack model such that it gives certain plaintexts. Say these ciphertexts set as C', then find all probable values of K5 and K6. Encrypt C' using these keys for additional MA half round say result is C". Now decrypt these set of ciphertexts in C" using three round IDEA decryption. Compare it with two and half round decryption of C' and compare these results. When there is match it means guessed value for K5 and K6 are correct else discard these wrong choices. The attack in overall recovers seventy three bits out of 128 bits of main key K. The positions of these bits in the 128 bit long main key are 10-15, 51-66, 67-82 and 106-121.

Biham targets simpler key scheduling algorithm in his attack by combining it with construction of linear equation involving the LSBs of many intermediate encryption values, using square like structures. The attack improves the previously known results significantly. The attack requires only 16 known PTs for attacking five round IDEA with time complexity of 2^{114} encryptions[1].

3.1.5 Weak Keys in IDEA

Keys are said to be weak when some of its subkeys are zero or one or they comprises of large number of zero bits. Weak keys turn the modular multiplication in IDEA encryption into a linear operation. Later it was explored that keys having runs of ones were also expressed as weak. 2^{51} differential weak keys were discovered by Daemen[14], *et al.* on IDEA cipher. He explained the test for membership of this weak class. This test requires only two encryptions and solution of sixteen non-linear Boolean equations, each equation having twelve variables. While Hawkes[15] searched out 2^{63} differential-linear weak keys for full cipher. Later on exploration, new weak key class was given by Wagner of size 2^{64}. He used a cryptanalytic technique called Boomerang attack. Adding to this, class of 2^{95} weak keys was discovered for 5-round IDEA on use of Boomerang membership test. These novel and advanced weak-key classes were built using Boomerang-style distinguisher. Weak-keys with zeros and ones can be fixed by using a constant which is XORed to all keys. While the problem of weak keys having runs of one's still remain unaccounted.

3.1.6 Biclique Attack

Biclique[18] attacks were introduced for hash function cryptanalysis as an extension to the initial structure technique, and later applied to block ciphers. In the biclique attack on block ciphers the full key space is partitioned into groups of keys, so that keys in a group can be efficiently tested in the meet-in-the-middle framework. The keyspace partition can be described in various ways. For permutation based key schedules as in IDEA we simply introduce three sets of key bits: K^b, K^f , and K^g. In a key group the value K^g is fixed (and hence enumerates the groups), and K^b and K^f take all possible values. Biclique attacks is a variant of MITM, it uses structure of biclique extend the number of rounds attacked by MITM. Biclique attack is 2^{18} times faster than brute force, an earlier intuition that this class of attacks only used for small speed-ups over brute force search is dismissed[19]. In this attack on block ciphers, groups of keys are generated by partitioning full key space, so that keys in a group can be efficiently tested in the MITM attack framework. Here, biclique is a set of internal states, constructed either in the first or in the last rounds of a cipher and mapped to each other by specifically chosen keys. The biclique approach to IDEA improves the cryptanalysis results on more rounds, and improves time and data complexities over existing attacks.

3.1.7 The combination of standard MITM with the keyless Biryukov-Demirci (BD) relation is used to obtain an attack on 6 round variant of IDEA. This starts after MA layer of second round. Here, data complexity is only 16 known plaintext and time complexity less than 2^{112} encryption[13]. In the attack computation of the same intermediate value V is performed in two separate ways, they divide the term equation in two sets, where the terms in the first set is calculated using only the plaintexts and set Kt of key bits, and terms of second set is calculated using ciphertexts and the set Kb of the key bits. For each guess of Kt, For the first set cryptanalyst calculates the exclusive OR of all terms in the equation saves it into table with a data structure based on hashing. After that using hit and trial approach for the sub-key Kb, cryptanalyst calculates the exclusive OR of all terms belonging to second set, and then look for a match in the table created. Various attack attempts on reduced round IDEA with chosen plaintext and required memory along with time complexity are as shown in Table 2.

3.1.8 *Data-Efficient Attacks on Reduced-round IDEA*

Initial attack on IDEA cipher has huge data requirement of 2^{127} which is as bad as average exhaustive search time for key. In 2009, Lai and Sun proposed first data efficient attack for 6 round. They reduced data requirement from 2^{64} to 2^{49} with time complexity from 2^{127} to $2^{112.1}$. Biham[20], *et al.* proposed innovative new data efficient attacks on reduced-round and the full IDEA cipher. They improved previous attack by Lai and Sun drastically in term of data complexity to just 16 plaintext data with almost same time complexity for 6 round IDEA. Biham[20], *et al.* could attack 6.5 rounds by increasing the data requirement from 16 to 1000 with time complexity of 2^{114}, by combining the improved MITM and the Biryukov-Demirci (BM) relation. By keyless version of BM relation used, they extended the attack to more number of rounds and with less data requirement. They explained best result for 7.5 rounds, in which they used MITM with slice and cut approach, with just 2^{63} data and 2^{114} time complexity they attacked 7.5 rounds. They extended the attack of full cipher using just 16 plaintext data with time complexity $2^{126.8}$, these results were very good in terms of data requirement. It is observed that there is a tradeoff between data complexity and time complexity, less the data complexity the more is time complexity.

Table 2. Comparison of time complexities of chosen plaintext attacks (CPA) on of IDEA cipher

Reference paper	Rounds	Type	No. C. Plaintext	Memory	Complexity
Biham & Shamir[4]	2	Differential	2^{10}	2^{23}	2^{42}
Knudsen & Rijmen[5]	2	Square-like	23	small	2^{64}
Biham & Shamir[4]	2.5	Differential	2^{10}	2^{96}	2^{106}
Hawkes[15]	3	Differential-linear	2^{29}	2^{16}	2^{44}
Demirci[6], *et al.*	3	Collision	2^{33}	2^{58}	2^{64}
Demirci[9]	3.5	Square-like	2^{34}	small	2^{82}
Demirci[6], *et al.*	3.5	Collision	2^{24}	2^{58}	2^{73}
Demirci[6], *et al.*	4	Collision	2^{24}	2^{58}	2^{89}
Biham[13], *et al.*	4.5	Impossible-differential	2^{64}	2^{32}	2^{112}
Biham[13], *et al.*	6	MITM BD-relation	16 KP	2^{25}	$2^{111.9}$
Biham[13], *et al.*	6.5	SaC MITM BD-relation	2^{23}	-	2^{113}
Biham[13], *et al.*	8.5	Biclique BD-relation	2^{52}	-	2^{126}

SaC-Splice-and-Cut, MIMT- Meet-in-the-Middle, CP/KP Chosen/Known Plaintext, BD Biryukov-Demirci, Time complexity measured in encryptions.

3.1.9 Schneier[21], *et al.* in their attack on IDEA exploited its key schedule since key schedule of IDEA is quite simple, sub keys are calculated by simply circularly shifting the original key 25 times in key scheduling algorithm. For 3-round IDEA, authors proposed a chosen key differential attack; it recovers 32 bits of key with 6 chosen plaintexts (PT). Out of these 6 PT, two are used with first key and the remaining four with the second key. In order to recover another 64 bits of key it needs 2^{17} chosen PT under third key. Rest of the 32 bits is recovered using brute forcedly trying all 2^{32}, choices. Schneier, *et al.* also proposed other variant of attack which is applicable on non reduced round IDEA with COA timing attack model. This attack enquires about 5x 2^{17}, related keys and it encrypts 10, 48,576 (2^{20}) random chosen unknown PT blocks with these related keys. But, the difficult part of this attack is to accurately measure the timings of these encryptions.

3.1.10 *Cryptanalysis of IDEA using SAT Solvers*

Nakahara[22], *et al.* investigates the power of SAT solvers in cryptanalysis. Their contributions are two-fold and are relevant to both theory and practice. Firstly, Authors explains how to convert the problem of cryptanalysis into system of equations and represent them efficiently into satisfiability (SAT) problem. Now the problem is encoded into SAT instance which solved using SAT solvers for solutions. The technique is efficient and generic. It can be applied on variety of symmetric block ciphers. This method can be used to automate the first step of algebraic attacks, i.e. the generation of a system of algebraic equations. Secondly, they mentions the scenarios wherein the SAT solvers are not suitable to attack cryptographic algorithms such as to find weak keys in symmetric key block ciphers and to compute the pre-images in hash functions. SAT solvers allow to find or prove the absence of, weak-key classes under both differential and linear attacks of full-round block ciphers based on IDEA. Generically, a weak key of a block cipher is a user key which leads to a non-random behavior of the cipher. Ideally, a block cipher should have no weak keys. For IDEA and related ciphers mentioned previously, a weak key causes some multiplicative subkeys to have value 0 or 1, turning multiplication into a linear operation. These multiplicative subkeys are mandatory inputs to a multiplication operation over GF ($2^{16} + 1$), where 0= 2^{16} by construction. Note that 2^{16} + 1 is a prime number. Authors have found classes of weak keys of block ciphers based on IDEA[4].

SAT solvers and cryptanalysis.

The Boolean satisfiability problem (SAT) was mentioned as a useful framework for cryptanalysis. The relation between input and output bits of a cryptographic algorithm can be expressed as a SAT problem which can then be supplied to a SAT-solver in order to recover the secret keys (e.g. key bits). The problem SAT can be solved by different types of solvers. Authors consider only complete SAT solvers, i.e. deterministic algorithms in which input is a given formula in conjunctive normal form (CNF) and output is always a satisfying valuation of its variables, in case such a valuation exists. Otherwise, the SAT solver outputs that the SAT instance is unsatisfiable. A CNF is a conjunction of clauses, where a clause is a disjunction of literals, and a literal is a propositional variable or its negation. Sometimes converting the problem into CNF is also quite challenging.

Finally their results shows weak keys of the full 8.5-round WIDEA-4 and WIDEA-8 block ciphers under differential and linear attacks. Standard techniques[22] to accelerate exhaustive search (brute force) on IDEA block cipher:

3.1.11. Standard Techniques[22] to Accelerate Exhaustive Search (Brute Force) on IDEA Block Cipher

(a) *The Early Abort Technique:* As soon as we observe a wrong ciphertext bit produced by a possible key, we can discard this possible key without computing the full ciphertext. Similarly we can discard whole class of possible keys which produces such a wrong ciphertext bit. E.g. in IDEA cipher last round called output transformation round; there are total four operations two multiplication and two additions each. On performing the first multiplication operation we get first 16 bits of ciphertext, if these bits are incorrect then no need to perform rest of the three operations out four operations. We can simply discard that particular possible key. Thus using early abort 75 per cent of efforts can be saved in output transformation round.

(b) *The Distributive Technique:* The distributive techniques for speed up of exhaustive trail is very much similar to the mathematical distributive law ab+ac=a(b+c), by taking common a out of parenthesis, here we compute a followed by b and again compute a followed by c which can be done like computing a once followed by (b+c) e.g. in the first round of IDEA cipher only first 96 bits are used for six subkeys, remaining 32 bits are not used in first round. First round operations does not use those last 32 bits of master key, so these operations can be performed once for all remaining choice of keys. Similarly we can apply this technique for decryption process also when few key bits are not used in last round of decryption process and can save lot of effort by not doing same thing multiple times.

(c) The distributive technique can be improved by clubbing it with MITM technique. When few key bits are not used in initial as well as last rounds operations, so adversary can perform these operations with all choices of key bits with MITM. e.g. IDEA cipher key bits 125-127 are not used in round 1,8,9 which allows us using distributive technique and MITM to reduce the exhaustive trial effort by 25 per cent, 3 out of 8.5 rounds.

4. CONCLUSION

This paper explains important attack on block ciphers. It explains working of IDEA cipher and its key scheduling algorithm. Here we give brief overview of PGP and how IDEA is used in it for providing confidentiality. It gives a review of the work related to the IDEA Block cipher and various attacks attempted over it. Additionally it shows how SAT solvers are useful in IDEA cryptanalysis. Use of SAT solvers and parallel implementation of SAT solvers may help to improve execution rounds within minimal complexity. In literature till 2015, there are attacks up to reduced round of IDEA with varying time and data complexities. Thus, there has not been a single attack that could break the practical security of IDEA cipher, which shows that IDEA is robust system for providing confidentiality and is still in use after 25 years of its inception.

REFERENCES

1. Biham, E.; Dunkelman, O.; Keller, N. & Shamir, A. New attacks on IDEA with at least 6 rounds. *Journal Cryptology*, 2015, **28**(2), 209-239. doi: 10.1007/s00145-013-9162-9
2. Lai, X.; Massey, J.L. & Murphy, S. April. Markov ciphers & differential cryptanalysis. In Workshop on the Theory and Application of of Cryptographic Techniques, 1991, 17-38. Springer Berlin Heidelberg. doi: 10.1007/3-540-46416-6_2
3. Philip R. Zimmermann.The Official PGP User's Guide. *MIT Press, Cambridge*, MA, USA, 1995. ISBN:0-262-74017-6
4. Biham, E. & Shamir, A. Differential cryptanalysis of DES-like cryptosystems. *Journal Cryptology*, 1991, **4**(1), 3-72. doi: 10.1007/BF00630563
5. Knudsen, J.D.L. & Rijmen, V. The block cipher SQUARE. In fast software encryption. *In* the Proceedings of 4th International Workshop, FSE'97, Haifa, Israel, January 1997. p. 149. doi: 10.1007/BFb0052343
6. Demirci, H.; Selçuk, A.A. & Türe, E. August. A new meet-in-the-middle attack on the IDEA block cipher. *In* International Workshop on Selected Areas in Cryptography, Springer Berlin Heidelberg. 2003, pp. 117-129. doi: 10.1007/978-3-540-24654-1_9
7. Meier, W. On the security of the IDEA block cipher. *In* Workshop on the Theory and Application of of Cryptographic Techniques, 1993, Springer Berlin

Heidelberg. pp. 371-385.
doi: 10.1007/3-540-48285-7_32

8. Borst, J. Differential-linear cryptanalysis of IDEA. ESAT–COSIC Technical Report, 1997.
9. Demirci, H. Square-like attacks on reduced rounds of IDEA. *In* International Workshop on Selected Areas in Cryptography, Springer Berlin Heidelberg, 2002, 147-159. doi: 10.1007/3-540-36492-7_11
10. Borisov, N.; Chew, M.; Johnson, R. & Wagner, D. Multiplicative differentials. *In* International Workshop on Fast Software Encryption, Springer Berlin Heidelberg, 2002, 17-33.doi: 10.1007/3-540-45661-9_2
11. Raddum, H. Cryptanalysis of IDEA-X/2. *In* International Workshop on Fast Software Encryption, Springer Berlin Heidelberg, 2003, pp. 1-8.
doi: 10.1007/978-3-540-39887-5_1
12. Demirci, H. & Selçuk, A. A meet-in-the-middle attack on 8-round AES. *In* International Workshop on Fast Software Encryption, Springer Berlin Heidelberg, 2008, pp. 116-126. doi: 10.1007/978-3-540-71039-4_7
13. Biham, E.; Dunkelman, O.; Keller, N. & Shamir, A. New data-efficient attacks on reduced-round IDEA. *In* IACR Cryptology ePrint Archive, 2011, 417.
14. Daemen, J.; Govaerts, R. & Vandewalle, J. Weak keys for IDEA. *In* Annual International Cryptology Conference, Springer Berlin Heidelberg, 1993, 224-231.
doi: 10.1007/3-540-48329-2_20
15. Hawkes, P. Differential-linear weak key classes of IDEA. *In* International Conference on the Theory and Applications of Cryptographic Techniques, Springer Berlin Heidelberg, 1998, 112-126. doi: 10.1007/BFb0054121
16. Wagner, D. The boomerang attack. *In* International Workshop on Fast Software Encryption, Springer Berlin Heidelberg, 1999, 156-170.
doi: 10.1007/3-540-48519-8_12
17. Biryukov, A.; Nakahara Jr, J.; Preneel, B. & Vandewalle, J. New weak-key classes of IDEA. *In* International Conference on Information and Communications Security, Springer Berlin Heidelberg, 2002, 315-326.
doi: 10.1007/3-540-36159-6_27
18. Bogdanov, A., Khovratovich, D. and Rechberger, C. Biclique cryptanalysis of the full AES. *In* International Conference on the Theory and Application of Cryptology and Information Security, Springer Berlin Heidelberg, 2011, 344-371. doi: 10.1007/978-3-642-25385-0_19
19. Khovratovich, D.; Leurent, G. & Rechberger, C. Narrow-Bicliques: cryptanalysis of full IDEA. *In* Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer Berlin Heidelberg, 2012, 392-410.
doi: 10.1007/978-3-642-29011-4_24
20. Biham, E.; Dunkelman, O. & Keller, N. A new attack on 6-round IDEA. *In* International Workshop on Fast Software Encryption, Springer Berlin Heidelberg, 2007, 211-224. doi: 10.1007/978-3-540-74619-5_14
21. Kelsey, J.; Schneier, B. & Wagner, D. Key-schedule cryptanalysis of idea, g-des, gost, safer, and triple-des. *In* Annual International Cryptology Conference, Springer Berlin Heidelberg, 1996, 237-251.
doi:10.1007/3-540-68697-5_19
22. Lafitte, F.; Nakahara Jr, J. & Van Heule, D. Applications of SAT solvers in cryptanalysis: Finding weak keys and preimages. *J. Satisfiability, Boolean Modeling Computation*, 2014, 1-25.

CONTRIBUTORS

Mr Harish Kumar Sahu is a part-time PhD scholar in the Department of Mathematics, IIT Delhi, India and currently working as Scientist ‘D’ in SAG /DRDO Delhi. He is an MTech in Computer Science and Engineering from IIT Delhi. His current research areas are applications of BDD and SMT in cryptanalysis, information security and android applications for data security.
In the current study, he has contribute Application of IDEA in PGP and SAT solver based cryptanalysis of IDEA.

Mr Vikas Jhadav is an scholar in the Department of Information Technology, Walchand College of Engineering, Sangli, MH, India. He is an MTech in Computer Science and Engineering from Walchand College of Engineering, Sangli. His current research area is information security.
In the current study, he has contribute survey, attacks on IDEA and block ciphers.

Prof. (Dr) Shefali Sonavane is an associate professor in Department of Information Technology, Walchand College of Engineering, Sangli, MH, India. She is a Phd in computer science & engg. Her current research areas are information security, steganography and theoretical computer science.
In the current study, she has contribute time complexities analysis of attacks on IDEA.

Prof. (Dr) R.K. Sharma is a professor in the Department of Mathematics, IIT Delhi, India. He is a PhD in Mathematics from IIT Delhi. His current research areas are algebra and cryptography.
In the current study, he has contribute meet in the middle (MITM) attack.