International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 23 (2015) pp 43395-43400

A Specialized Assault Adjacent To The Aes And Its Function To System Implementations

M. Navaneetha Krishnan

Research Scholar Department of CSE Manonmaniamsundarnar University Tirunelveli, India. mnksjce@gmail. com

Dr. R. Ravi

Professor & Head Department of IT Francis Xavier Engineeing College Tirunelveli, India.
directorresearch@francisxavier. ac. in

Abstract
Algebraic side channel attack is a major technique to solve the ciphers and the various side channel attacks. A side channel attack is a type of attack based on information gathered from the application of a cryptosystem, relatively than brute force. Common classes of side channel attack comprise, Timing attack, Power monitoring attack, Acoustic cryptanalysis, Differential fault analysis, Data permanence. Efforts to breakdown a cryptosystem by deceptive or forcingindividuals with genuine access are not classically called side-channel attacks. It is problematic to gross into explanation the specificassets of the targeted cryptographic algorithms. Depending on the kind of solver the time complexity is fairly high afterseeing the error-tolerant attack situations. To discover a novelmethod that would reduce the impact of the general solver and to decrease the time complexity of side channel attacks, a new algorithm named Rijndael algorithm is being proposed. An AES algorithm to defend the side channel attacks against the attacker module. Thus in AES, the encrypted text will be sent into the several rounds of encryption with the key. So that the decryption of file will be happened with the same key. A new approach has been proposed to overcome the side channel attacks and various prevention techniques has been implemented to overcome the attacks. The prevention techniques implemented are used in login pages to prevent bruteforce attack. Then the admin viewable keys are reencrypted using double encryption techniques and the various attacks are stimulated, by which they are prevented from attacks.

Keywords— Side channel attack, Rijindael algorithm, AES algorithm, General solvers, Algebraic side channel attack.

Introduction

Security evaluation of cryptographic algorithms can be considered in two aspects. From the point of view of mathematical security such as differential, linear, algebraic cryptanalysis and their variants. And their implementations can be considered in the point of view of physical with side channel-analysis techniques where physical leakages from the target devices, such as execution time power consumption and electromagnetic emissions are exploited to break the algorithms. Embedded systems are most vulnerable to SCAs as attackers often have direct physical access. Typical SCA techniques include simple power analysis, differential power analysis, correlation power analysis, mutual information analysis, template analysis, stochastic SCA, side-channel cube analysis, algebraic side-channel collision analysis and algebraic SCA. All these attacks exploit some model of the physical leakages to be compared with actual measurements. Assumptions regarding the model lead to two important classes of SCAs: profiled and non-profiled.

A profiling phase where the adversary is provided with a training device under test (TRDUT) that allows to characterize physicalleakages (in order to obtain a precise leakage model); and an online exploitation phase where the attack is mounted against a similar target device under test (TADUT) to perform a secret key extraction. Non-profiled SCA requires only the latter phase and assumes some (less precise) leakage model, typically obtained from engineering intuition. With respect to the key recovery procedure, SCAs fall into two categories: divide-and-conquer SCA (which provide distinguishers for small key chunks that are then combined, e. g., using key enumeration) and analytical SCA (which recover the entire key at once, e. g., by solving systems of equations).

The SCAs can be divided into the four types. Analytical SCA is currently a very active area in the crypto community. Traditional SCAs exploit a divide-and-conquer strategy and recover several pieces of a secret key independently. For analytical SCA, both the cipher and the leakages are represented with algebraic equations and the full secret key is recovered at once by solving these equations with different strategies. Since leakages of more rounds can be utilized, this attack has less measurement complexity than traditional SCAs. To attack the software implementation of AES on 8-bit microcontroller, collision-based SCA is combined with algebraic cryptanalysis. The attack is named algebraic side-channel collision analysis (ASCCA).

In ASCCA, the adversary detects the internal collisions (if the values of two intermediate states are equal) in two AES rounds by comparing the patterns of two sections of the power traces and then converts them into equations. The F4 Gröbner basis-based algorithm in MAGMA solver is used to solve the equations. Under known plaintext scenario, ASCCA only requires five power traces to recover the master key of AES. This attack is independent of the leakage model. In ASCA, template attack is used to deduce the Hamming weight (HW) or the accurate value of intermediate states.

This can be done by detecting the external collisions between the targeted power trace in TADUT with the template power trace in TRDUT. The algebraic technique is used to represent both the cipher and the deductions. A SAT solver, ZChaff, is used to recover the secret key. Compared with ASCCA and other SCA techniques, ASCA can exploit the side-channel leakages in all cipher rounds and can recover the key with a single trace even when both the plaintexts and ciphertexts are unknown. ASCA works well on the software under the Hamming weight leakage model (HWLM). Recently, it has also been successfully applied to the hardware implementation of AES on a 65nm ASIC, under the template leakage model (TLM) with a single power trace.

This work studies the impact of representation dependence, leakage dependence and cipher dependence on ASCA. Also the impact of algebraic immunity to the resistance of block ciphers against ASCA is studied. The original ASCAs assume that the correct deduction on the Hamming weight (HW), or the accurate value of intermediates states, can be profiled from analyzing the side-channel leakages. In practice, it was observed that noise is the main issue for robust ASCA. Because of this, multiple deductions have to be obtained from the leakage and utilized in the attack. To mitigate this issue, two types of solutions are provided. One solution is to group the deductions together into sets, then convert them into algebraic equations. A SAT solver is normally used to recover the secret key. In this approach, there are many variants, such as multiple deductions-based ASCA (MDASCA) in COSADE 2012 and improved ASCA (IASCA) in HOST 2012. The other solution is to include the imprecise deductions in the equation set and to deal with these imprecisions via an optimizer (e. g., the SCIP solver.

This technique is denoted as Tolerant ASCA (TASCA) in CHES 2010 and in Eprint 2012/092. In CHES 2012, TASCA is further modified to cope with the different probabilities of multiple deductions named probabilistic TASCA (Prob-TASCA). Prob-TASCA can regain some information lost in other attacks. In summary, most existing ASCAs adopt a general, sometimes off-the-shelf equation solver (e. g., the F4 Gröbner basis-based algorithms in MAGMA solver, SAT solver, mixed integer programming solver). The main feature of the general solver approach is its unique technique, which can be applied to different cryptographic algorithms. The main drawback is to take into account the specialized structures or properties of the specific cryptographic algorithms. The results differ depending on the type of solver used, and the time complexity.

When there is too much noise and the deduction sets are too large, there exist too many solutions for the equation system. Equationset, the general equation solver might give a satisfied or optimized solution but not the correct one, which reduces the success rate. If both the plaintext and ciphertext are included in the equation set, the output solution should be correct but the solver may not give the correct solution in a reasonable amount of time. It is critical to find a new tactic that would minimize the impact of the solver and reduce the time complexity of existing ASCAs on AES, especially when considering the error-tolerant attack scenario. The main idea is inspired by the simple power attack technique in ICISC 02 and the low data complexity attack technique in CRYPTO 11.

The work in utilized the incomplete diffusion feature in the AES key expansion to recover the secret key of AES with a single power trace. The work is utilized as customized solver approach instead of the general equation solver to solve the equations of Round-Reduced AES. It is interesting to exploit the incomplete diffusion feature in the AES encryption procedure and utilize a specialized approach (construct a customized or specialized solver) instead of the general equation solver to improve ASCA. Since there are more leakages in the AES encryption procedure, the attack might work under unknown plaintext andcipher text scenario. Meanwhile, as the incomplete diffusion feature is considered, this specialized attack may achieve better performance than existing ASCAs. We name our technique incomplete diffusion of analytical side-channel analysis.

Related Work

Nicolas Veyrat-Charvillon, Benut Gerard, Francois-Xavier Standaert[1] rank estimation algorithm is used to identify the key sizes for symmetric cryptography which are usually standardized. This leads to an uncomfortable situation, where the security of an implementation can be anywhere between enumerable values of the full key size.

Charles Bouillaguet, Patrick Derbez, and Pierre-Alain Fouque[2] To demonstrate the strengh of the tools, this ispermitted to automatically realizethe new attacks on round-reduced AES through very low data complexity, and thus to find the enhanced attacks on the AES. Algorithmsachievement the algebraically simple byte-oriented arrangement of the AES.

S. Chari, J. Rao, and P. Rohatgi[3] Execution of AES, this is not amenable to methods such as SPA and DPA, can simply be shattered using a template attacks with a single sample. The achievement of these attacks in such obligingcircumstances is due to the manner in which noise inside each sample is handled.

ItaiDinur and Adi Shamir[4] The conceptof leakage attacks is on iterated block ciphers, in which the attacker can discover physical searching, power measurement, or any other type of side channel. The unique cube attack needsparticularly clean data, however the information provided by side channel attacks can be quite noisy.

X. J. Zhao et al[5] The Algebraic side-channel attack (ASCA) is a powerful cryptanalysis techniquewich is highly different from conventional side-channel attacks. To enhance ASCA, they propose a general method, called Multiple Deductions-based ASCA (MDASCA), to cope the multiple deductions produced by erroneous measurements or intrusions.

Erfan Aghaee Majid Rahimi Hamed Yuse[6] On removing the quadratic equations, identifying iterated chosen plaintexts, and cube iteration to progress the SCCA on block ciphers. The Side Channel Cube Attack (SCCA) is a sympathetic of Algebraic Side Channel Attack (ASCA) consisting of theoretical and practical phases.

Proposed algorithm

Here the proposed AES with Rijndael algorithm. First, we analyse the encryption features with the advanced AES

algorithm. Typical SCA techniques include simple power analysis (SPA), differential power analysis (DPA), correlation power analysis (CPA), mutual information analysis (MIA), template analysis (TA), stochastic SCA (SSCA), side-channel cube analysis (SCCA), algebraic side-channel collision analysis (ASCCA) and algebraic SCA (ASCA). All these attacks exploit some model of the physical leakages to be compared with actual measurements.

Improved Advanced Techniques

A. Substitue Bytes Method

This stage (known as SubBytes) is basically a table lookup using a (16×16) matrix of byte values named as s-box. This matrix containsof the potential amalgamations of an 8 bit sequence (28 = 16 × 16 = 256). Though, the s-box is not the only random permutation of these standards and there is a well distinct technique for generating the s-box tables. This is not too concerned how the s-boxes are made up and can simply take them as table lookups. The matrix that gets operated in the process of encryption is known as state. More concern is focused on how this matrix is effected in each round. In this particularround each byte is mapped to a novel byte in the subsequent way: the leftmost nibble requires a precise row and the right nibble requires a column of the S box. This is formerly used to update the state matrix. The Inverse is used to substitute byte transformation (known as InvSubBytes) which makes usage of an inverse s-box. The s-box considered to be strong is known for cryptanalytic attacks. Precisely, the Rijndael designers hunted a scheme that has a low connection among input bits and output bits, and the assets that the output cannot be defined as a humble mathematical function of the input. In addition to this, the s-box takes no fixed points (s-box(a) = a) and no conflicting fixed points (s-box(a) =−a) where −a is the bitwise compliment of a. The s-box is essential to be an invertible and if decryption is to be made possible (Is-box[s-box(a)]= a) it should not be its self inverse i. e.
s-box(a) 6= Is-box(a)

B. Shift Row Transformation Method

This stage (known as ShiftRows) is a simple permutationand nothing else. The working hierarchy of this method is:

- The first row of the byte in the state is not altered.
- The second row of the byte is shifted 1 bytes to the left in a circular style.
- The third row of the byte is shifted 2 bytes to the left in a circular style.
- The fourth row of the byte is shifted 3 bytes to the left in a circular style.

The Inverse of the Shift Row transformation (known as InvShiftRows) performs these circular shifts in an opposite manner for the last three rows (the first row which is unaltered to begin with). Thus thisprocess may not seem to do much but if you think nearly how the bytes are well-arrangedinsidethe state then it can be realised to have an impact. The state is preserved as an array of four byte columns, such that the first column actually represents bytes 1, 2, 3 and 4. A one byte shift is hence a linear distance of four bytes. The transformation also ensures that the four bytes of one column are ranged out to four different columns.

C. Mix Column Transformation

This stage (known as MixColumn) is fundamentally a substitution but it sortsthe use of arithmetic of GF(28). Each column is activated on separately. Every byte of a column is plotted to a new value which is a function of all the four bytes in the column. The transformation can be determined by the subsequent matrix multiplication on state.

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} t_{0,0} & t_{0,1} & t_{0,2} & t_{0,3} \\ t_{1,0} & t_{1,1} & t_{1,2} & t_{1,3} \\ t_{2,0} & t_{2,1} & t_{2,2} & t_{2,3} \\ t_{3,0} & t_{3,1} & t_{3,2} & t_{3,3} \end{bmatrix} = \begin{bmatrix} t'_{0,0} & t'_{0,1} & t'_{0,2} & t'_{0,3} \\ t'_{1,0} & t'_{1,1} & t'_{1,2} & t'_{1,3} \\ t'_{2,0} & t'_{2,1} & t'_{2,2} & t'_{2,3} \\ t'_{3,0} & t'_{3,1} & t'_{3,2} & t'_{3,3} \end{bmatrix}$$

Each component of the product matrix is the sum of product of elements of one row and one column. In this case the separate additions and multiplications are done in GF(28). The MixColumns transformation of a single column is thus expressed as j ($0 \le j \le 3$) of state can be expressed as:

$$t'_{0,b} = (2 \cdot t_{0,b}) \oplus (3 \cdot t_{1,b}) \oplus t_{2,b} \oplus t_{3,b}$$

$$t'_{1,b} = t_{0,b} \oplus (2 \cdot t_{1,b}) \oplus (3 \cdot t_{2,b}) \oplus t_{3,b}$$

$$t'_{2,b} = t_{0,b} \oplus t_{1,b} \oplus (2 \cdot t_{2,b}) \oplus (3 \cdot t_{3,b})$$

$$t'_{3,b} = (3 \cdot t_{0,b}) \oplus t_{1,b} \oplus t_{2,b} \oplus (2 \cdot t_{3,b})$$

where • signifies multiplication over the finite field GF(28). The InvMixColumns is defined by the following matrix multiplication

$$\begin{bmatrix} 0C & 0F & 0G & 09 \\ 09 & 0C & 0F & 0G \\ 0G & 09 & 0C & 0F \\ 0F & 0G & 09 & 0C \end{bmatrix} \begin{bmatrix} t_{0,0} & t_{0,1} & t_{0,2} & t_{0,3} \\ t_{1,0} & t_{1,1} & t_{1,2} & t_{1,3} \\ t_{2,0} & t_{2,1} & t_{2,2} & t_{2,3} \\ t_{3,0} & t_{3,1} & t_{3,2} & t_{3,3} \end{bmatrix} = \begin{bmatrix} t'_{0,0} & t'_{0,1} & t'_{0,2} & t'_{0,3} \\ t'_{1,0} & t'_{1,1} & t'_{1,2} & t'_{1,3} \\ t'_{2,0} & t'_{2,1} & t'_{2,2} & t'_{2,3} \\ t'_{3,0} & t'_{3,1} & t'_{3,2} & t'_{3,3} \end{bmatrix}$$

If the label of these A and A−1 individually and the label state themix columns operation as S and subsequently as S',
This is defined as:

$$AS = S'$$

therefore

$$A{-}1S' = A{-}1AS = S$$

System Design

Typical SCA techniques include simple power analysis (SPA), differential power analysis (DPA), correlation power analysis (CPA), mutual informationanalysis (MIA), template analysis (TA), stochastic SCA (SSCA), side-channel cube analysis (SCCA), algebraic side-channel collision analysis (ASCCA) and algebraic SCA (ASCA). All these attacks exploit some model of the physical leakages to be compared with actual measurements.

In Figure [1]The sender sends the file which is prevented from all the side channel attacks. The key is generated by the sender which is encrypted along with the file and transferred to the receiver. The admin thus accepts and sends a mail which is used to decrypt the file. This makes the AES encryption more secure from the side channel attacks.

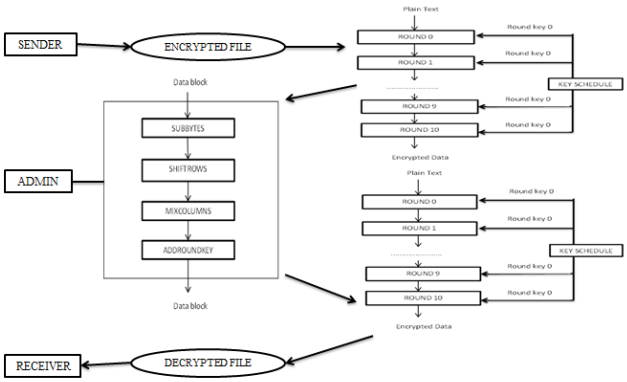

Fig. 1. **System Architecture.**

A. Description of AES

AES is a symmetric block cipher. This means that it uses the same key for both encryption and decryption. However, AES is quite different from DES in various ways. The Rijndael algorithm allows a variety of block and key sizes and not just 64 bit and 56 bit of DES block and key size. The block and key can infact be chosen independently from 128, 160, 192, 224, 256 bits and need not be the same. However, the AES standard states that the algorithm can only accept a block size of 128 bits and a choice of three keys-128, 192, 256 bits. Depending on the version used, the name of the standard is modified to AES-128, AES-192 or AES-256 respectively. The main difference between AES and DES is such that, DES is not a feistel structure. In a feistel structure, half of the data block is used to modify the other half of the data block and then the halves are swapped. In this case the entire data block is processed in parallel during each round using substitutions and permutations.

Rijndael was designed to have the following characteristics:

- Resistance against all known attacks.
- Speed and code compactness on a wide range of platforms.
- Design Simplicity.

B. Implementation of AES

KeyExpansions—round keys are obtained from the cipher key by means ofRijndael's key schedule. AES needs a distinct 128-bit key for each round plus one more.

InitialRound

AddRoundKey—In the AddRoundKey step, the subkey is joined through the state. On behalf of each round, a subkey is found after the main key byRijndael's key schedule; everysubkey is of the identical size as per the state.

SubBytes—In the SubBytes step, every byte a_{i, j}which is in the state matrix is swappedby a SubByte S(a_{i, j}) by means of an 8-bit substitution box, the Rijndael S-box. Thisprocesshelps in finding the non-linearity in the cipher.

ShiftRows—TheShiftRows step functions on the rows of the state; it cyclically shifts the required bytes in every row by a definite offset value. In AES, the first row remainsunaffected. Every byte of the second row is shifted one place to the left.

MixColumns—In the MixColumns step, four bytes of every column are mutual by an invertible linear transformation. The MixColumns function comprises of four bytes as input and four bytes as output, where each and every input byte affects all four output bytes. Both techniques of ShiftRows, MixColumns provides diffusion in the cipher.

AddRoundKey
Final Round (no MixColumns)
SubBytes
ShiftRows
AddRoundKey.

C. Divide the States and Leaks in Each AES Round

State group mapping. The calculation procedure involved in state group mapping consists of one state group in βi from the state group in αi as the stategroup mapping, which can be expressed as

$$\alpha i \rightarrow \beta i$$

Leak group. According to the software implementation ofAES, there are 21 leaks analyzed in one state groupmapping, which can form a leak group and can be denoted as

$$L\alpha i \rightarrow \beta i$$

D. Conquer the States and Leaks in Each AES Round

Let E(P) denote the entropy of the state P. Let x denote a state byte, D(x) denote the deductionset on the valueof L(x), x denote one possible candidate of L(x), andI s(x, D(x)) denote the function of D(x). Let Pi, Ri, Ci, Di, Qi denote the 128-bit input, round-key, then the output of CR, output of SD and the final output of the i-th round, respectively.

Let pij, rij, cij, dij, qij, denote j-th byte of Pi, Ri, Ci, Di, Qi ($0 \le i \le 9$, $0 \le j \le 15$), presents the equation which can be calculated {qi0, qi1, qi2, qi3} from pi and the round key Ri for the first nine rounds ($0 \le i \le 8$). As to the final round (i=9). function to judge whether x is in thededuction set D(x). This will describe how to analyzefor the state group αj by analyzing $L\alpha j \rightarrow \beta j$ and $D\alpha j \rightarrow \beta j$

$$(0 \le j \le 3).$$

E. Search for the Master Key

Recovering any round key is equal to the recovery of the master key. The candidates of the master key by the technique of brute-forcesearch of Ri for each round and then the method of intersection is used among the candidates in multiple rounds to compute the final search of the master key. The complexity while computing the intersection varies with the candidate size in different rounds, which is affordable with small size and intensive with large.

Discussions

The main objective of the project is to prevent the file being attacked by the attacker, by which we prevent the various side channel attacks to the user. Thus by using the AES algorithm we prevent various attacks and share the data from sender to the receiver using proper requirement methods. Thus the AES prove to be the best algorithm to prevent the Side channel attacks from being attacks.

This proposes a new technique named Incomplete Diffusion Analytical Side-channel Analysis (IDASCA). The main analysis of the incomplete diffusion in one AES round is described as the core of IDASCA on AES, which is mainly composed of three steps: states and leaks in each AES round, state from leaks in each AES round and search for the master key of AES.

$$q^i_0 = 02.\ S[p^i_0+r^i_0]+03.\ S[p^i_5+r^i_5]+S[p^i_{10}+r^i_{10}]+S[p^i_{15}+r^i_{15}]$$

$$q^i_1 = S[p^i_0+r^i_0]+02.\ S[p^i_5+r^i_5]+03.\ S[p^i_{10}+r^i_{10}]+S[p^i_{15}+r^i_{15}]$$

$$q^i_2 = S[p^i_0+r^i_0]+S[p^i_5+r^i_5]+02.\ S[p^i_{10}+r^i_{10}]+03.\ S[p^i_{15}+r^i_{15}]$$

$$q^i_3 = 03.\ S[p^i_0+r^i_0]+S[p^i_5+r^i_5]+S[p^i_{10}+r^i_{10}]+02.\ S[p^i_{15}+r^i_{15}]$$

Results

A. Attack Without Errors

IDASCA can work well in both known plaintext and unknown plaintext/ciphertext scenarios. The output of CR and the round output in the i-th round can be recovered by analyzing the 84 leaks (excluding the 16 leaks in loading the round key Ki) in that round. This is been reduced to a greater extend to analyze the i+1-th round, since the final output of the i-th round, also an input of CR in the i+1-th round, is unknown, it is not possible to extract the round-key from any single round between Round 2 and 9.

B. Attack With Errors of Fixed Deduction Size

In this, the two different values of the deduction size μ are considered, the same as the prob-TASCA in CHES 2012. The first is μ = 2, where they conduct 100 attacks in the first 9 rounds of AES and the average time obtained is 1 second. The results explain that ρd = 80% and ρk = 80% which means that the master key of AES can be recovered by attacking the first round under known plaintext scenario, or unknown plaintext/ciphertext scenario. When μ ≥ 4, under unknown plaintext/ciphertext scenario, the effort to handle and reduce E(Ri+1) is unaffordable (296) and IDASCA is prevented using a single power trace of one plaintext. If power traces of multiple plaintext/ciphertext pairs are used, since new unknown 256-bit variables are introduced in each new pair, the complexity of IDASCA would be greater than single pair attack and IDASCA is also prevented.

C. Attack With Errors and Dynamic Deduction Size

In practical, an adaptive technique can apply the dynamic μ approach and select the highest n HW deductions foreach leak, when the sum of these probabilities are over a fixed threshold T. When IDASCA is applied on AES for this scenario, the attack complexity is much lower than the attack with fixed μ. Thus the threshold and and value of μ is not fixed.

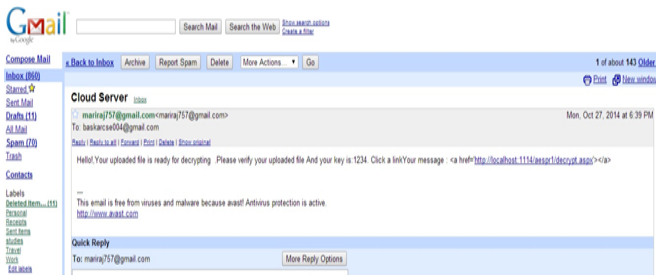

Fig 2. Key Transformation

Fig 3. Key Acquisition

Fig 4. File Decryption

Conclusion

There are several interesting problems for further research. First is how to utilize the different probabilities of multiple deductions to generate the rank for each state candidate (four bytes). Second is how to apply the key ranking enumeration and estimation algorithms to IDASCA, and exploit the rank to estimate the remaining key strength in ASCA more accurately.

References

[1] K. Jung, "Text information extraction in images and video: A survey, "PatternRecognit., vol. 37, no. 5, pp. 977-997, May 2004.

[2] J. Liang, D. Doermann, and H. Li, "Camera-based analysis of text and documents: A survey, " Int. J. Document Anal. Recognit., vol. 7, nos. 2-3, pp. 84-104, 2005.

[3] J. Zhang and R. Kasturi, "Extraction of text objects in video documents: Recentprogress, " in Proc. 8th IAPR Int. Workshop Document Anal. Syst., Sep. 2008, pp. 5-17.

[4] S. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "ICDAR 2003 robustreading competitions, " in Proc. Int. Conf. Document Anal. Recognit., 2003, pp. 682-687.

[5] S. Lucas, "Icdar 2005 text locating competition results, " in Proc. Int. Conf. DocumentAnal. Recognit., 2005, pp. 80-84.

[6] A. Shahab, F. Shafait, and A. Dengel, "ICDAR 2011 robust reading competition challenge2: Reading text in scene images, " in Proc. Int. Conf. Document Anal. Recognit., 2011, pp. 1491-1496.

[7] X. Chen and A. Yuille, "Detecting and reading text in natural scenes, " in Proc. IEEEComput. Soc. Conf. Comput. Vis. Pattern Recognit., 2004, pp. 366-373.

[8] X. Chen and A. Yuille, "A time-efficient cascade for real-time object detection: Withapplications for the visually impaired, " in Proc. IEEE Conf. Comput. Vis. PatternRecognit., Workshops, Jun. 2005, pp. 1-8.

[9] C. Yi and Y. Tian, "Text string detection from natural scenes by structure-based partitionand grouping, " IEEE Trans. Image Process., vol. 20, no. 9, pp. 2594-2605, Sep. 2011.

[10] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations innatural images, " in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2012, pp. 1083-1090.