An R-SIFT Image Matching Intelligent Algorithm Applied to Hardware Trojan Detection

Chen Sun¹, Pujiang Liang², Lingling Li²*, Jingjing Ma³, Licheng Jiao² and Fang Liu²

¹Science and Technology on Reliability Physics and Application of Electronic Component Laboratory, Guangzhou 511370, China
²Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center for Intelligent Perception and Computation, Joint International Research Laboratory of Intelligent Perception and Computation, School of Artificial Intelligence, Xidian University, Xi’an, 710071 China

*Corresponding author email: llli@xidian.edu.cn

Abstract. Malicious modification, deletion or addition of some modules in the integrated circuits (ICs) will cause the chip to be attacked, which is called Hardware Trojans (HTs). Reverse engineering (RE) is a classic destructive method and widely used in HTs detection. However, RE is very time-consuming and error prone. In this paper, based on RE, we propose the R-SIFT algorithm to match image and reformulate the Trojan detection problem as change detection problem. For the R-SIFT algorithm, the ratio of exponentially weighted averages (Roewa) operator is introduced into the scale-invariant feature transform (SIFT) algorithm. Experiments show that the proposed method has 7 to 56 times more matching points than the original SIFT can be effectively applied to HTs detection.

Keywords: Trojan detection; R-SIFT; Image matching.

1. Introduction

In recent years, integrated circuits are often the core component of a device, mainly responsible for control and storage functions. If there is a problem with the integrated circuit, it will have a great impact on the use of the device. Therefore, it is very important to ensure the safety and reliability of integrated circuits. Once the integrated circuit is planted with hardware Trojans, the entire device will have potential use risks [1]. In order to solve this problem, there are many methods for detecting hardware Trojans in integrated circuits.

Hardware Trojan detection can be roughly divided into two categories, destructive and non-destructive. Non-destructive methods mainly include logic testing and side channel analysis methods [2]. Reverse engineering (RE) is a destructive method. The authors in [3] have investigated many different reverse engineering methods and compared their advantages and disadvantages. However, most of the non-destructive methods require a lot of tests on the circuit and data under different conditions, and a large amount of operating data needs to be analyzed [4]. Moreover, non-destructive detection methods are difficult to detect hidden Trojans. In contrast, destructive methods can more directly find abnormal parts in the actual circuit [5].

Reverse engineering is a widely used destructive method with 5 main steps: decapsulation, delayering, imaging, annotation, and schematic creation. The first three steps are mainly to obtain the actual image of the integrated circuit, and the latter two steps are mainly to obtain the design layout from the actual
produced layout images [6]. In fact, the latter two steps are time-consuming and some hidden Trojans will be missed when generating the design layout. Therefore, these two steps are unnecessary and can be replaced by image comparison algorithms. Using the image match method to compare the produced circuit with the designed circuit layout, we can directly find the difference. Therefore, the image match method can be directly and efficiently applied to hardware Trojan detection.

The SIFT algorithm is proposed by Professor David G. Lowe of Canada [7]. The SIFT algorithm has multi-scale characteristics and direction invariance, and its feature points are distinguishable and robust. In this paper, we proposed the R-SIFT by introducing Roewa operator into the SIFT algorithm and applied it to hardware Trojan detection. First, we illustrated some hardware features and common Trojans, then we explained the content of the R-SIFT algorithm, and finally verified our method through experiments and analyzed the experimental results. The main contributions of this paper are as follows:

Based on the Reverse Engineering, we propose to use image matching method to solve the problem of Trojan detection. As previously introduced, the use of image matching methods for hardware Trojan detection can replace the last two steps in Reverse Engineering, which can reduce time consumption. We introduced the Roewa operator into the SIFT algorithm and proposed the R-SIFT algorithm. The R-SIFT algorithm can greatly reduce the interference of noise points in the image on the image matching result. Because there are many noise points in the actual circuit layout image, and the characteristic structures of the ICs are similar, R-SIFT can effectively process the ICs image data.

We use the actually produced circuit chip and the design layout of the corresponding chip to simulate the method we proposed. The results show that our proposed method can accurately match chip circuit images with different resolutions and different rotation angles. The final detection results also show that our method is of great help to the detection results of various types of hardware Trojans.

The rest of paper is organized as follows. Section 2 presents several common types of hardware Trojans and the proposed R-SIFT algorithm. In section 3, the introduction of data and experimental results are provided. Finally, the conclusion is given in Section 4.

2. The Proposed Method

![Common types of hardware Trojans.](image)

Figure 1. Common types of hardware Trojans.

Hardware Trojans in the actual circuit chip layout usually have three situations: Trojan Addition, Trojan Deletion and Trojan Parametric. Trojan Addition refers to the addition of some circuit components to the original design layout. Trojan Deletion refers to the deletion of some circuit components in the original design layout. Trojan Parametric refers to the replacement of some circuit components in the original design layout. The display results of these three different hardware Trojan forms in the picture are shown in Fig. 1 [6].

In this paper, we use the Roewa operator to replace the gradient calculation method in the original SIFT algorithm, which is called R-SIFT. The original SIFT algorithm uses Gaussian difference function as the basis for gradient calculation. This method is unstable when dealing with outliers (such as noise and speckles). Roewa operator is a calculation method based on neighborhood information statistics. By calculating and averaging neighborhood information, outliers can be effectively avoided, and a stable false alarm rate can be maintained in a uniform area with high and low reflectivity. The flow chart of R-SIFT feature point matching algorithm is shown in Fig. 2.
By constructing a multi-scale space, the detected feature points can be scale-invariant. Gaussian pyramids can be constructed by convolving the original image with Gaussian functions of different scales. The difference of Gaussian (DoG) pyramid can be obtained by subtracting the two adjacent Gaussian filtered images of the same octave in the Gaussian pyramid as follows:

\[
D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) \ast I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma)
\]

where \(x\) and \(y\) represent the horizontal and vertical position coordinates of the image, \(\sigma\) is the blur factor of the Gaussian filter, \(I(x, y)\) represents the original image, \(k\) is a scale factor, \(L(x, y, \sigma)\) represents the content in the Gaussian pyramid, \(G(x, y, \sigma)\) is the Gaussian kernel function, and its content is as follows:

\[
G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}
\]

By constructing a DoG pyramid, feature points can be detected in a multi-scale space. In DoG, by comparing each pixel with its 26 surrounding points (8 adjacent points in the same scale and 18 points in the surrounding scale) to determine whether it is an extreme value, we can determine whether the pixel is a feature point. By removing the feature points that have an excessively large ratio of curvature in the horizontal and vertical directions, edge effects can be eliminated and the effectiveness of the feature points can be further improved. Finally, the position of the feature point in the original image can be obtained by interpolation.

In order to make the feature descriptor have direction invariance, it is necessary to arrange the direction of the feature points. The calculation formula of the gradient and direction in the original SIFT algorithm is as follows:

\[
m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}
\]

\[
\theta(x, y) = \arctan \left( \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right)
\]

Different from the original SIFT algorithm, we use the Roewa operator method to calculate the gradient and direction as the following formula:

\[
m(x, y) = \sqrt{I(\sigma, x)^2 + I(\sigma, y)^2}
\]

\[
\theta(x, y) = \arctan \left( \frac{I(\sigma, y)}{I(\sigma, x)} \right)
\]

Where \(m(x, y)\) represents the magnitude of the gradient at the point, and \(\theta(x, y)\) represents its direction. \(I(\sigma, x)\) and \(I(\sigma, y)\) respectively represent the first derivative in the x and y directions at the point, and the calculation method is as follows:

\[
I(\sigma, x) = \log \left( \frac{M_1(\sigma, x)}{M_2(\sigma, x)} \right)
\]
\[ I(\sigma, y) = \log \left( \frac{M_z(\sigma, y)}{M_z(\sigma, y)} \right) \] \tag{9}

\[ M_z(\sigma, x) = \int_{-R}^{R} \int_{-R}^{R} I(a + x, b + y) \times e^{-\frac{|x|^2}{\sigma}} \, dx \, dy \]

\[ M_z(\sigma, y) = \int_{-R}^{R} \int_{-R}^{R} I(a + x, b + y) \times e^{-\frac{|y|^2}{\sigma}} \, dx \, dy \]

\[ M_z(\sigma, y) = \int_{-R}^{R} \int_{-R}^{R} I(a + x, b + y) \times e^{-\frac{|x|^2}{\sigma}} \, dx \, dy \]

\[ M_z(\sigma, y) = \int_{-R}^{R} \int_{-R}^{R} I(a + x, b + y) \times e^{-\frac{|y|^2}{\sigma}} \, dx \, dy \] \tag{10}

It can be seen that the method of using the Roewa operator can make full use of the neighborhood information, thereby avoiding the influence of noise points. By calculating the magnitude and direction of the gradient, a histogram of the gradient direction can be established. The 360° directions are divided into 36 groups, and the direction with the largest directional derivative is taken as the main direction, and the other direction derivatives that are greater than 80% of the main direction are taken as the auxiliary direction.

When generating feature descriptors, the neighborhood around the feature point needs to be divided into 4*4 subregions. The histogram of gradient directions of each sub-region contains 8 directions. Therefore, for a key point, its feature descriptor is a vector containing 128 (4*4*8) elements. Finally, the feature is resized to one dimension to enhance its stability for subsequent feature point matching. By calculating the Euclidean distance of the feature descriptors of the two feature points, the similarity of the two feature points can be obtained.

3. Experiment

In order to show that the R-SIFT algorithm proposed in this paper can effectively detect hardware Trojans, two chip circuit diagrams with different resolutions are used in the experiments. The design layout is the layout data of the M3 layer of gdsii and the display file in drf format provided by the design company, and the cadence software is used for display. As shown in Fig. 3, (a) is an image transformed from design layout, and (b) is an image of the circuit. The resolution of the design layout image is 929*1899, and the resolution of the actual circuit image is 1280*1040. It can be seen that the color difference between the actual circuit and the design layout is obvious, and it will be affected by illumination and angle, which brings great challenges to the matching algorithm. The SIFT algorithm and R-SIFT algorithm were compared experimentally on these two pictures.

**Figure. 3.** Experimental data.
As shown in Fig. 4, it can be clearly seen that the R-SIFT algorithm has accurate matching results. Due to the introduction of the Roewa operator, the feature points detected by the R-SIFT algorithm are more accurate, avoiding the influence of some noise points. In contrast, the result of the SIFT algorithm is very messy, with many mismatch points.

Table 1 further illustrates the difference between the results of the two algorithms. Due to the introduction of different methods of calculating the gradient and direction, even when the number of feature points is similar, the correct matching points of the R-SIFT algorithm is higher than that of the SIFT algorithm. Moreover, while improving the high matching accuracy rate, the matching time did not increase significantly. The appearance of the components in the hardware circuit diagram is similar, so their features are also similar. Using neighborhood methods such as the Roewa operator to calculate the gradient can avoid the impact of similar features on the matching results.

Table 2. Comparison of experimental results on similar scales.

|       | Feature points in image 1 | Feature points in image 2 | Match points | Matching time(s) |
|-------|---------------------------|---------------------------|--------------|------------------|
| SIFT  | 25911                     | 19846                     | 44           | 49               |
| R-SIFT| 25911                     | 19846                     | 326          | 51               |

4. Discussion

In order to further illustrate the robustness of our proposed algorithm, we conducted experiments with multiple scales and rotations. As we know, the SIFT algorithm has good multi-scale characteristics and rotation invariance. In this experiment, the actual circuit diagram image was scaled to 0.6 times the original. Fig. 5 and Table 2 show the experimental results when the scales of the two images are quite different. It can be seen that on the basis of maintaining the SIFT algorithm, the R-SIFT algorithm is better than the SIFT algorithm at different scales.

Table 2. Comparison of experimental results on similar scales.

|       | Feature points in image 1 | Feature points in image 2 | Match points | Matching time(s) |
|-------|---------------------------|---------------------------|--------------|------------------|
| SIFT  | 25911                     | 14273                     | 9            | 41               |
| R-SIFT| 25911                     | 14273                     | 273          | 38               |
Fig. 6 and Table 3 show the experimental results when the rotation angle of the two images differs greatly. It can be seen intuitively from Fig. 6 that the proposed method has better matching results. Furthermore, the data in Table 3 also shows that R-SIFT improves the accuracy of matching without increasing the calculation time. It can be seen from the above experiments that our proposed method can increase the number of correct matching points without increasing the calculation time, thereby improving the matching accuracy. More importantly, the proposed method has multi-scale characteristics and rotation invariance, and has achieved good results in image matching of hardware circuit boards. It further illustrates that the proposed method can be further applied to the Trojans detection of hardware circuits.

| Feature points in image 1 | Feature points in image 2 | Match points | Matching time(s) |
|---------------------------|---------------------------|--------------|-----------------|
| SIFT                      | 25911                     | 17340        | 5               | 60              |
| R-SIFT                    | 25911                     | 17340        | 284             | 49              |

Figure 5. Comparison of experimental results with large scale differences.

Figure 6. Comparison of experimental results under different rotation angles.

Table 3. Comparison of experimental results on similar scales.
5. Conclusion

In this paper, an R-SIFT algorithm is proposed for Hardware Trojan Detection. The R-SIFT algorithm uses the Roewa operator to replace the directional gradient calculation in the original SIFT algorithm, reducing the influence of noise points. Because of the decrease of noise points in the circuit layout, the experimental results show that the R-SIFT algorithm has 7 to 56 times more matching points than the original SIFT and can effectively solve the hardware circuit registration problem. In the future work, we will further use deep learning methods to detect hardware Trojans after matching.

Acknowledgements

This work was supported in part by Project supported the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (No.61621005), Science and Technology Program of Guangzhou, China (No. 201904010210), the National Natural Science Foundation of China (No. 61801124, 61906150, 62076192, U1701267, 61836009), the New Think-tank of Department of Education In Shaanxi Province of China (No. 20JT021).

References

[1] R. Karri, J. Rajendran, and K. Rosenfeld, “Trojan taxonomy,” in Introduction to Hardware Security and Trust, M. Tehranipoor and C. Wang, Eds. Springer New York, 2012, pp. 325–338.
[2] G. Zarrinchian and M. S. Zamani, “Latch-based structure: a High resolution and self-reference technique for hardware trojan detection,” in IEEE Transactions on Computers, vol. 66, no. 1, pp. 100-113, 1 January. 2017.
[3] S. E. Quadir, J. Chen, D. Forte, N. Asadizanjani, S. Shahbazmohamadi and L Wang et al. “A survey on chip to system reverse engineering” JETC, 13(1):6:1±6:34, 2016.
[4] Zhao Yiqiang, Yang Song, He Jiaji, et al. Hardware Trojan Detection Method Based on Principal Component Analysis [J]. Journal of Huazhong University of Science and Technology (Natural Science Edition), 2015 (8): 66-69.
[5] ZHANG Peng, WANG Xincheng, ZHOU Qing. Chip Hardware Trojan Detection based on Electromagnetic Radiation Signal Analysis[J].
[6] Bao, Chongxi, Domenic Forte, and Ankur Srivastava. "On application of one-class SVM to reverse engineering-based hardware Trojan detection." Fifteenth International Symposium on Quality Electronic Design. IEEE, 2014.
[7] David G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints” International Journal of Computer Vision, 60, 2(2004), pp.91-110.