A fast-blind IP watermark detection scheme based on position fuzzification

Weihong Huang*

College of Computer Science and Electronic Engineering,
Hunan University,
Changsha, Hunan, China
and
College of Computer Science and Engineering,
Hunan University of Science and Technology,
Xiangtan, Hunan 411201, China
Email: whhuang@hnu.edu.cn
*Corresponding author

Renfa Li

College of Computer Science and Electronic Engineering,
Hunan University,
Changsha, Hunan, China
Email: 840233730@qq.com

Jianbo Xu

College of Computer Science and Engineering,
Hunan University of Science and Technology,
Xiangtan, Hunan 411201, China
Email: jbxu63@163.com

Yin Huang

College of Software,
Xiamen University of Technology,
Xiamen, Fujian 361024, China
Email: yinhuang131@163.com

Yong Sheng

College of Computer Science and Engineering,
Hunan University of Science and Technology,
Xiangtan, Hunan 411201, China
Email: 564359147@qq.com

Abstract: The rapid advancement in internet of things (IoTs) has created new demands for intellectual property (IP) protection. Existing zero-knowledge-based IP watermark detection schemes have low performance in real-time detection. To address this issue, we propose a position fuzzification-based IP watermark detection scheme mainly aiming to reduce the rounds of inquiry. Firstly, a random sequence is produced by chaos system to scramble all the resource positions of an IP design. Then watermark positions are fuzzified to make the scramble algorithm irreversible. A verifier can achieve zero-knowledge proof of IP ownership by one round of inquiry. Experiments show that the proposed scheme can greatly reduce computational complexity during blind IP watermark detection and enhance performance in real-time watermark detection.

Keywords: IP watermark; position fuzzification; zero knowledge protocol; blind detection.

Reference to this paper should be made as follows: Huang, W., Li, R., Xu, J., Huang, Y. and Sheng, Y. (2019) 'A fast-blind IP watermark detection scheme based on position fuzzification', *Int. J. Embedded Systems*, Vol. 11, No. 1, pp.94–105.

Biographical notes: Weihong Huang is PhD student in School of Computer Science and Electronic Engineering, Hunan University. He is currently a Lecturer in School of Computer Science and Engineering, Hunan University of Science and Technology. His research interests include information security, embedded system, and hardware IP protection.

Renfa Li is a Full Professor of Computer Science and Electronic Engineering at the Hunan University, China. He is the Director of the Key Laboratory for Embedded and Network Computing of Hunan Province, China. He is also an expert committee member of National Supercomputing Center in Changsha, China. His major research includes embedded computing, parallel computing, and cyber-physical systems. He is a senior member of IEEE, and a senior member of ACM.

Jianbo Xu is a Professor in School of Computer Science and Engineering, Hunan University of Science and Technology. He is a senior member of China Computer Federation. His current research interests include wireless sensor network, embedded system, and information security.

Yin Huang holds an MS from the School of Computer Science and Engineering, Hunan University of Science and Technology. He is currently a lecturer in college of software, Xiamen University of Technology. His research interests include hardware security and IP protection.

Yong Sheng holds an MS from the School of Computer Science and Engineering, Hunan University of Science and Technology. His research interests include hardware security and IP protection.

1 Introduction

With the rise of mobile intelligent terminals and internet of things (IOTs) in recent years, integrated circuits (IC) have been rapidly developing (Fernandes et al., 2017; Liang et al., 2017; Yampolskiy et al., 2014). Field programmable gate array (FPGA) has become the mainstream of IC design. Reuse technology in the embedded system products has been widely used. It brings about many conveniences, but also creates big challenges in IP infringements (Colombier et al., 2016; Bossuet and Torres, 2017; McDonald et al., 2016; Ngo et al., 2017). To address this problem, researchers have proposed to use digital watermark in IP protection, namely IP watermarking technology (Sengupta and Bhadauria, 2017). This technology can identify copyright by hiding watermarks in an IP core. When infringement happens, it is easy to extract watermarks in the IP core to resolve the copyright dispute.

Owing to the particularity of IP design, many effective watermarking schemes have been proposed for IP ownership protection (Jung et al., 2015; Zhang and Liu, 2017; Saha and Sur-Kolay, 2007). Lach et al. (2006) firstly proposed a physical IP watermarking technology based on lookup table (LUT). The watermarks are hidden into unused LUTs. The path routing is adjusted to improve the security of watermark positions. Zhang et al. (2012) proposed a method that embeds watermarks in the configuration file directly. It makes watermark embedding and extraction convenient. Liang et al. (2011b) proposed an FSM-based IP watermarking method by adding watermarks into the maximum delay state set of a particular circuit. Fan (2008) inserted the watermark circuit into original IP design and proposed five methods to detect watermarks. Cui and Chang (2012), Chang and Cui (2010) and Cui et al. (2011) proposed a testable constraint-based watermarking scheme at behavioural design level. It packets the logical grid and randomly selects the testable modules to embed watermarks. Liang et al. (2011a) proposed an IP watermarking scheme based on multiple scan chains in sequential logic circuits. It has good performance in reliability and resource overhead. To enhance the security of IP watermarking techniques, Schmid et al. (2012) proposed a robust watermarking algorithm based on LUT structure. The watermarks can be well concealed, making the removal attacks difficult. But the drawbacks are that the power and time delay will be affected. Castillo et al. (2007) proposed to host the signature within memory structures or combinational logic that are part of the system. But the module for watermark detection should be added in the design, causing large hardware overhead. The logic is easily removed as well. Qu (2002) introduced the third party to authenticate IP ownership. The inserted watermarks are divided into public watermark and private watermark. The private watermark can only be detected by several legal users. This algorithm can address the difficulty in watermark authentication to some degree. Jain et al. (2003) embedded watermarks by altering the last bits of some time constraints. It is a zero-overhead IP watermarking algorithm, but watermark detection is not convenient. If the measured time delay is not accurate, the watermark detection will fail.

Lots of previous researches utilise the LUT structure of FPGA for watermark insertion. It is a convenient method, but the resource occupancy, delay and power will be affected by the inserted watermarks. So, it should be considered to make full use of FPGA structure, compress the watermark size and make it recoverable in extraction.

There are several standards to measure IP watermarking schemes (Liang et al., 2016b; Lach et al., 1999; Raju et al., 2009), including function interference, security, robustness, traceability, etc. In traditional IP watermarking schemes,

watermark detection always requires sensitive watermark positions. It is sensitive, if leaked, since it is probable for an adversary to remove these watermarks. Thus, some watermarked IP designs may lose protection. Therefore, secure detection of IP watermark becomes a research hotspot. At present, blind IP watermarking detection requires many inquiring rounds to produce results, causing low performance in real-time detection. This paper introduces position fuzzification to reduce the number of inquiry rounds in detection and improve real-time performance in detection.

2 Relative works

Traditional watermark detection algorithm reveals the real information about the watermark to the verifier, as shown in Figure 1. It is not safe because the verifier may leak the watermark information to others. To address this issue, researchers have proposed a zero-knowledge proof-based IP watermark detection algorithm. It will prove the existence of the watermark without leaking sensitive information about the watermark. However, the algorithm is constrained by the reversibility of the position scrambling algorithm. So, the watermark verification should be divided into two parts and execute multi-rounds of verification to ensure security. The zero-knowledge proof-based watermark detection involves a complex position scrambling algorithm, which usually use the image scrambling algorithm, such as Hibert curve, Z curve and Arnold transformation. The claimer firstly scrambles the resource positions of IP design. The scrambled design and the watermark positions are sent to the verifier, who will perform the watermark detection. If the scrambling parameters are not public, the verifier cannot derive the original watermark positions. Figure 2 describes the above procedure.

Figure 1 The traditional IP watermarking procedure (see online version for colours)

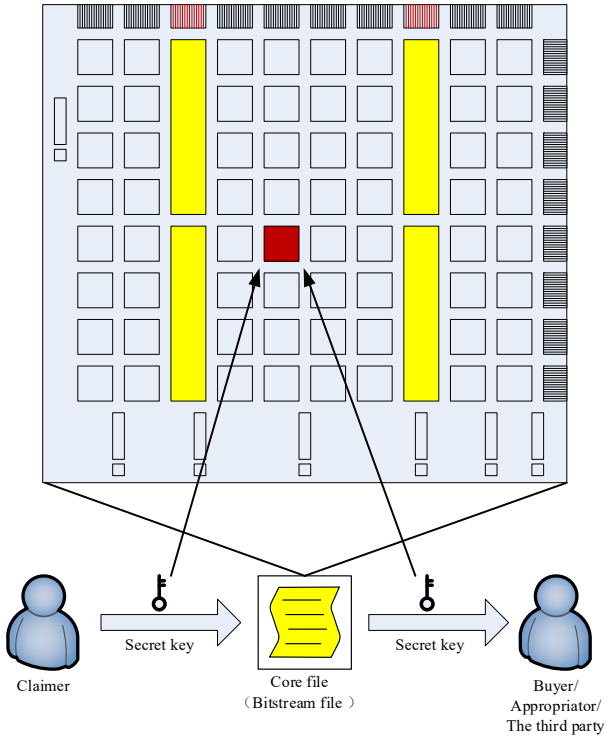
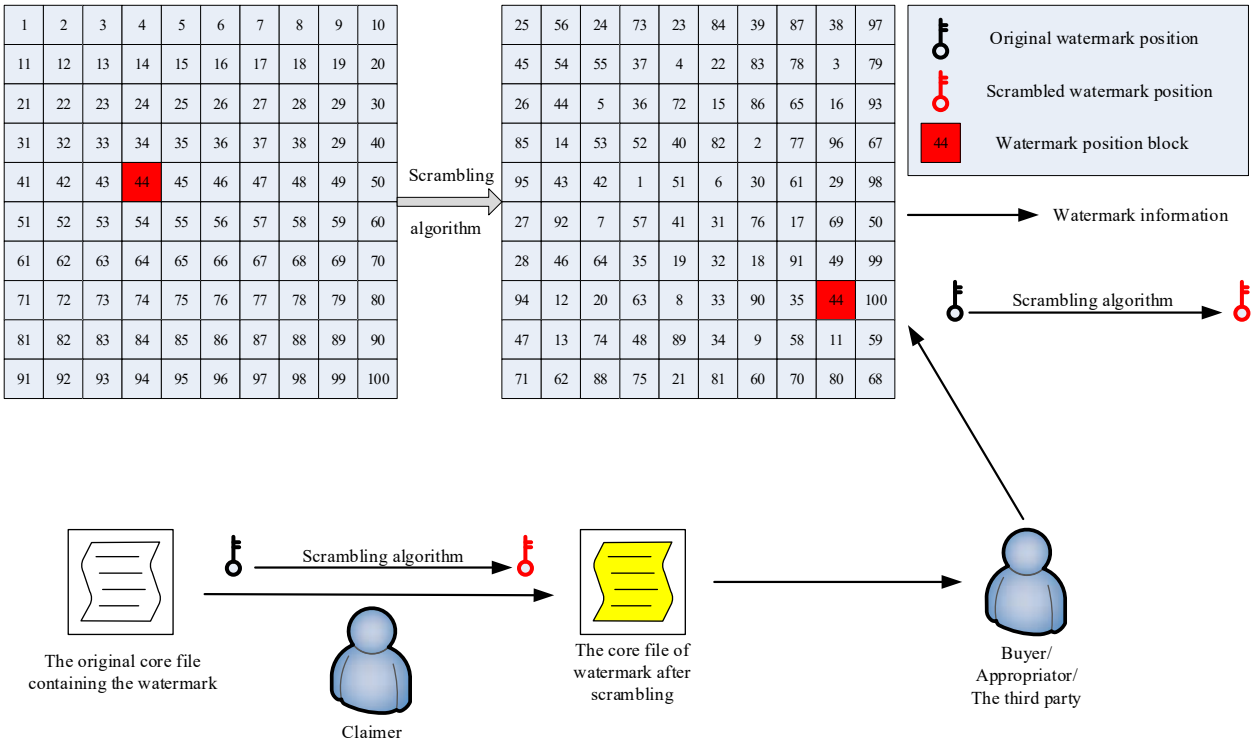


Figure 2 The flow of blind watermark detection (see online version for colours)



The claimer claims the IP ownership to the verifier. After that the verifier will send an inquiry request to the claimer. The inquiry usually includes a set of information to verify the identity. The proof procedure will not leak the sensitive information of the watermarks, which is realised by obfuscation. The claimer will reply the inquiry and send the response to the verifier. The verifier will convince the response or not. If not, the verifier can continue to send the inquiry request until he convinces the response of the claimer. The multi-rounds of inquiry reduce the probability of cheating. If one of the inquiry results is failed, the authentication is not successful. On this basis, some researchers proposed an improved authentication scheme, such as FFS, which sends several queries at a time to reduce the communication overhead of the authentication. Guillon Quisquater identity authentication protocol uses RSA to reduce the information exchange in each inquiry. Schnorr identity authentication protocol uses the difficulty in calculating the discrete logarithm. The complexity in computation and communication is much lower. The watermark detection should ensure the security of the watermark positions. So, the zero-knowledge proof can be used in watermark detection.

2.1 IP watermarking blind detection technology base on zero knowledge protocol

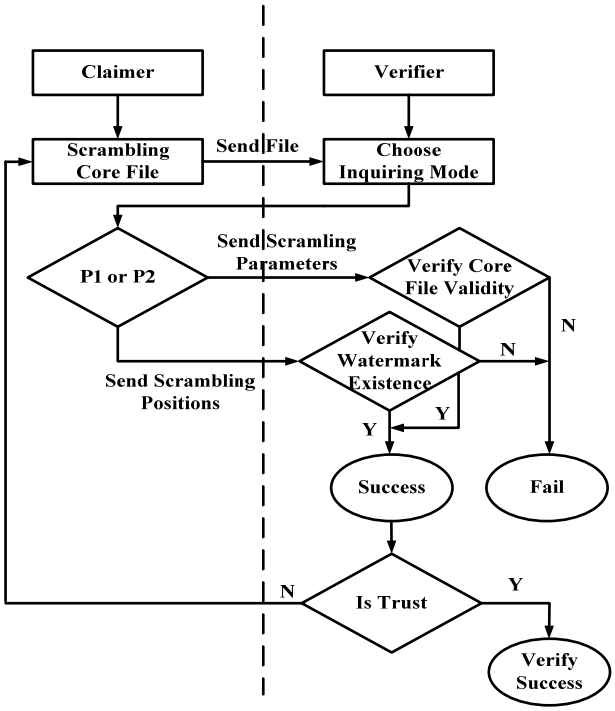
The traditional watermarking extraction needs to provide real watermark locations. It poses a great threat to security of watermarks, because it cannot ensure whether the watermark locations will be leaked. Saha and Sur-Kolay (2012) firstly proposed an IP watermarking blind detection based on zero knowledge protocol. It does not require the intervention of third party during watermarking detection, and it is a public IP watermarking detection algorithm. The owner scrambles the IP core to confuse real watermark locations. It only provides scrambled watermark locations. So, the actual watermark locations are not leaked in detection.

The zero-knowledge-based blind IP watermarking detection is a process with N inquiry rounds. Figure 3 describes the detailed steps. Saha divided the blind detection into the following two parts:

P1 Verify the effectiveness of IP core: the owner sends the scrambled core to verifier which may be forged by the sender. The verifier needs to verify whether the scrambled core is true. Verification method requires the owner to send scrambling parameters to the verifier, and the verifier needs re-scramble original core fie. If both the scrambled cores are same, the core is valid.

P2 Verify the existence of watermarks: it verifies whether the received IP includes a watermark proving IP copyright. It requires the owner to send scrambled watermark locations to the verifier, and the verifier uses the locations to extract watermarks in the scrambled core. Because the verifier knows nothing about scrambling parameters, he cannot reverse the original watermark positions.

Figure 3 Flow chart of core watermark blind detection based on zero knowledge protocol



Saha conceals real watermark locations successfully using the above method. The security of this program can be measured by $1 - (1/2)^n$. n indicates the number of inquiry rounds. When n is close to infinity, the security of this scheme is the best. But IP core needs to re-scramble in each round of inquiring which limits the real-time performance of watermarking detection.

Many researchers have extended this algorithm based on Saha's idea. Zhang et al. (2013) proposed a blind detection scheme based on chaos theory. This scheme uses a chaotic system to generate random values which are used to control whether the positions are swapped during procedure of scrambling IP core. It has good performance in scrambling watermark locations, but it also needs n inquiry rounds to ensure security. Liang et al. (2016a) proposed a blind IP watermarking detection scheme using Hibert curve to scramble IP core. The identity authentication protocol is used in watermarking detection for better security. Similarly, this scheme also needs multiple inquiry rounds and has low real-time performance.

By analysis, low real-time performance is mainly caused by the reversible position scrambling algorithm. This means that watermark detection needs two parts of verification. So, it is necessary to design an irreversible scrambling algorithm to make the detection rounds to be constant.

2.2 Kent chaotic mapping system

Chaotic mapping system (Wang et al., 2004; Munir et al., 2007) is generally used to solve the power system, fractal, chaos and other classic models of complex system behaviour. Kent mapping is a kind of chaotic system. It is superior to traditional logistic map in randomness and ergodicity. The mapping relationship is:

$$F(x)=\begin{cases} x/S, & x\in(0,S] \\ (1-x)/(1-S), & x\in(S,1) \end{cases} \tag{1}$$

Chaotic system is extremely sensitive to initial values. In formula (1), S represents the control parameters of chaotic system. When x, $S \in (0, 1)$, regardless of how small distance number is chosen, the initial two track spacings will increase by index rate over time so that it cannot be predicted. Many researchers also confirm that the sequence generated by initial condition x_0 under the Kent mapping has a great uniformity. In this paper, Kent chaotic mapping system is used to generate random positions scrambling correspondence.

2.3 Fuzzy mapping function

This paper proposes to use fuzzy concept to further confuse the real watermarking locations. The confused locations are no longer reversible. Therefore, it has higher security. The fuzzification process needs a fuzzy mapping function, which is used to generate a fuzzy radius. This function is not fixed, it is chosen by claimer. In this paper, we use a modified sigmoid function as the fuzzy function. Unipolar sigmoid function is also called single-ended S-shaped function, because it can map the real field to the range of (–1, 1). It is usually used as a middle level excitation function in the neural network model. The mapping relationship is:

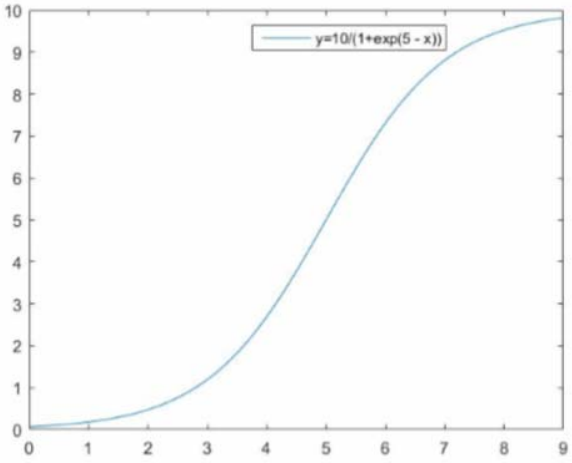
$$S(x)=\frac{1}{1+\exp(-ax)} \tag{2}$$

where a is the tilt parameter. It can affect the tilt degree of function. When $a = 1$ and $x > 5$, the dependent variable is infinitely close to 1. In this paper, the sigmoid function is only used to generate a range of fuzzy positions. It also can be regarded as a probability function to generate a number between (0, 1). After repeated experiments, the sigmoid function needs to be changed slightly, and the changed mapping relationship is:

$$S(x,y)=\frac{10}{1+\exp\left(-\left(fmod\left(Lut_{x,y},10\right)-5\right)\right)} \tag{3}$$
$$x\in[0,m];\ y\in[0,n]$$

where $Lut_{x,y}$ represents the value stored in corresponding LUT of core file. *fmod* represents a modulo operation of floating point data. The value calculated by formula (3) is the effective radius of position fuzzy, namely fuzzy radius. It is a very important parameter in inquiring process.

Figure 4 Relationship between dependent and independent variables of sigmoid function (see online version for colours)



3 Fast blind detection of IP watermarking based on fuzzy position

3.1 IP watermarking fast blind detection interaction model

This paper proposes a fast-blind detection algorithm based on fuzzy position. It realises the blind detection process in constant rounds. Figure 5 depicts an interactive model of the proposed scheme. It requires the claimer and verifier to shake hands three times and only needs one inquiring round in detection. The inquiring process is as follows.

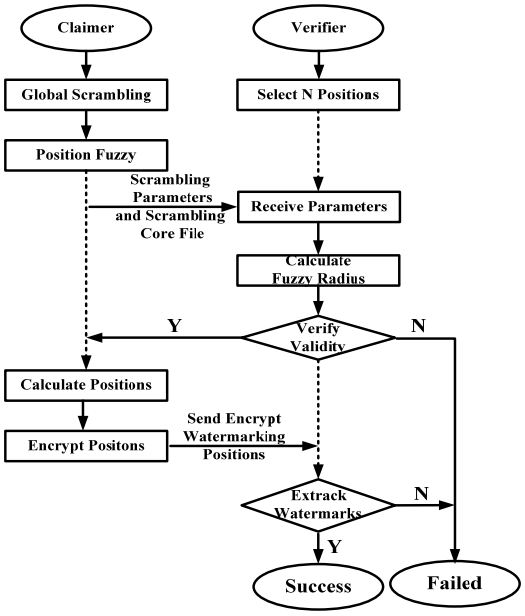
Step 1 The claimer firstly utilises Kent chaotic system to generate scrambling sequence. This sequence is used to scramble the position of LUTs in the IP core. Secondly, claimer executes position fuzzification process to make the position scrambling algorithm irreversible. Lastly, the claimer sends scrambling parameters and scrambled IP core to the verifier.

Step 2 The verifier selects n positions of the original IP core to verify the effectiveness of the scrambled IP core. If all of n positions are matched successfully, the scrambled IP core is valid. The verifier sends a success message to the claimer for further verification.

Step 3 Once the claimer receives a successful message, scrambled watermark locations are calculated. These locations are encrypted by and sent to the verifier.

Step 4 After receiving encrypted watermarking locations, the verifier firstly decrypts the locations, then verifies the existence of watermarks in the scrambled IP core.

Figure 5 The interaction model of fast blind watermark detection algorithm



3.2 Fast blind IP watermarking detection algorithm

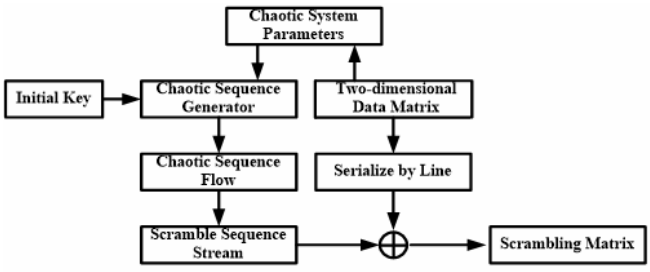
3.2.1 Initialisation phase

Firstly, we separate data from the LUT of the core file and use it to initialise a two-dimensional data matrix $B[m][n]$. Secondly, we prepare some protocol parameters which are needed in the inquiring process. Verifier V selects t positions from the two-dimensional data matrix $B[m][n]$ and use $L_v = \{l_1, l_2, l_3, \ldots, l_t\}$ to indicate. Verifier V prepares key pair. The public key is (e_v, n_v). The private key is d_v, and the encryption mode can choose RSA or other asymmetric encryption algorithms.

3.2.2 Global position scrambling phase

In this phase, we use resorted corresponding positions generated by Kent chaotic mapping to scramble the core. Figure 6 shows the global scrambling process.

Figure 6 Global scrambling flowchart



The detailed global scrambling process is as follows.

Step 1 Scan two-dimensional data matrix $B[m][n]$ in line and the result is converted into a one-dimensional sequence $P = \{p_1, p_2, p_3, \ldots, p_{m\times n}\}$ with the length of $m \times n$.

Step 2 Accumulate data from sequence P, use *sum* to indicate the result. Calculating control parameter S by formula (4), and calculate the iteration number of Kent chaotic system K. $MD5(B)$ indicates MD5 hash value of the original core. It is used to resist against replay attack of blind detection.

$$S = sum/m \times n \times 2^{16} + MD5(B)/2^{128} \tag{4}$$

$$K = 10^3 + \mathrm{mod}(sum, 10^3) \tag{5}$$

Step 3 The claimer randomly selects $x_0 \in (0, 1)$ as the initial key of formula (4), iterating formula (4) K times to eliminate the impact of transient effects.

Step 4 Continue iterating $m \times n$ to generate a chaotic sequence flow $T = \{t_1, t_2, t_3, \ldots, t_{m\times n}\}$, and order this sequence to obtain a sequence $T' = \{t_1', t_2', \ldots, t_{m\times n}'\}$. We can get a sequence $T_N = \{T_{N_1}, T_{N_2}, \ldots, T_{N_{m\times n}}\}$ based on the change of the elements positions between sequence T and sequence T'.

Step 5 Sequence T_N is regarded as the result sequence P after scrambling denoted by $p_i' = p_{T_{N_i}}$ (i = 1, 2, 3, …, $m \times n$). We construct the two-dimensional matrix $B'[m][n]$ using sequence P' according to the size of $m \times n$.

Upon executing the above steps, the global scrambling process is completed. The scrambled IP core is represented by $B'[m][n]$. With global scrambling parameter is (S, K, x_0). Global scrambling process is described as $B'[m][n] = G(S, K, x_0, B[m][n])$.

3.2.3 Fuzzy position

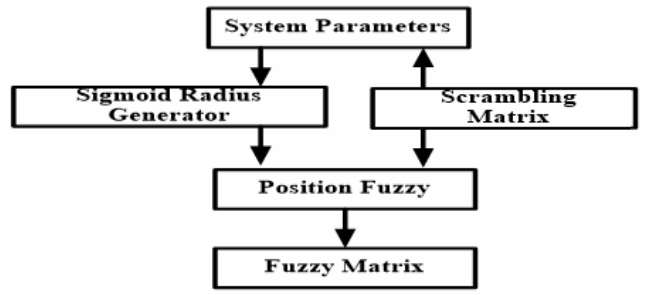
The essence of position fuzzification is to further confuse real position of watermarks, break the reversibility of the global scrambling algorithm and guarantee the security of the watermark location. This process is shown in Figure 7. The detailed process is as follows.

Step 1 Select the position of each data block from $B'[m][n]$, substitute the fuzzy radius formula (3) to calculate. The results are fuzzy radius $R[m][n]$. We use $R(Lut_{x,y})$ to indicate fuzzy radius formula.

Step 2 Choose every position (i, j) $i \in (0, m)$ $j \in (0, n)$ to make a fuzzy transformation $f(i, j\ r)[i][j] = (i', j')$. Set the scrambled position as the centre point and search for unoccupied idle position within the radius of $R[i][j]$ to switch. If there are no unoccupied idle positions within the range, no switching is performed.

Step 3 The position of each block data in the core file is selected in turn to make a fuzzy transformation and the core file after fuzzy transformation is $B''[m][n]$. Formula $F(m, n, R(B'[m][n])) = B''[m][n]$ is utilised to describe this process.

Figure 7 Position fuzzification flow chart



After performing the above steps, position fuzzification is completed. Because of adding random selection operation during the position fuzzification process, the verifier cannot reverse the original watermark positions from scrambled positions.

3.2.4 Watermark inquiring process

After global scrambling and position fuzzification, the claimer gets a scrambled IP core $B'[m][n]$ and scrambling parameters (S, K, x_0, $R(Lut_{x,y})$). Then claimer and verifier interact three times to complete IP core blind detection. Figure 8 describes watermarking inquiring phase interaction diagram, and the specific inquiring process is as follows:

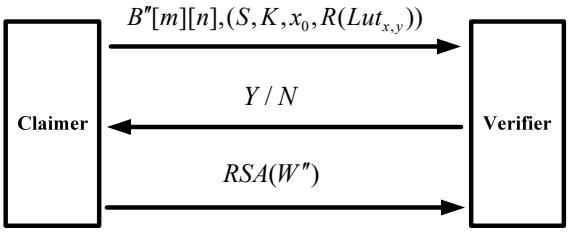
Step 1 The claimer sends scrambled IP core $B''[m][n]$ and scrambling parameters (S, K, x_0, $R(Lut_{x,y})$) to the verifier.

Step 2 After the verifier receives messages, positions $L_v = \{l_1, l_2, \ldots, l_t\}$ is selected to calculate whether $M(L_v, R(G(S, K, x_0, B[l_{t_x}][l_{t_y}])) = L_v''$ is true. L_v'' indicates L_v after scrambling. The fuzzy matching operation process $M((x, y), r)$ is as follows. Firstly, a fuzzy radius $R(G(S, K, x_0, B[l_{t_x}][l_{t_y}]))$ is calculated. Secondly, search for a target within this radius. If searching is successful, message YES sent to the claimer. If the searching fails, a message of failed validation is returned to the claimer.

Step 3 If the claimer receives the YES message, global scrambled positions W' of the original watermarking positions are calculated. We then calculate fuzzy positions W'' from W'. (e_v, n_v) is regarded as the public key of RSA encryption. The encrypted positions $RSA(W'')$ is sent to the verifier.

Step 4 After the verifier receives $RAA(W'')$, fuzzy positions W'' is decrypted by private key d_v. The verifier uses W'' to extract watermarks from $B''[m][n]$.

Figure 8 Watermark questioning stage interaction diagram



3.3 Security analysis

3.3.1 Replay attack

Replay attacks usually happen in the field of digital watermarks. General solution is to add time stamps during watermarks embedding process. This paper presents a method that sets the MD5 hash value of the core file as the input to the chaotic system. If the counterfeit claimer embedded new watermarks in the IP core during blind detection, the MD5 hash value of the original IP core would be quite different and the scrambled effects will also be different. The verifier can distinguish whether the core file is a replay attack according to the MD5 hash values of the original core and the core received from the sender. Figure 9 describes a schematic diagram of this algorithm against replay attack.

3.3.2 Position scrambling security

The purpose of position scrambling is to break the reversibility of the global scrambling, reduce the rounds of inquiring and improve real-time performance. Therefore, the complexity of position scrambling is directly related to the security of this protocol.

The size of the LUT array in core file is $m \times n$. Fuzzy radius is $r \in (0, m + n/2)$. Ignoring the global scrambling process, every position has $4r^2$ different choices, i.e., there are a total of $(4r^2)^{mn}$ possibilities. Huge position possibilities are sufficient to achieve secure protocol requiring.

Figure 9 Schematic diagram of resist replay attack (see online version for colours)

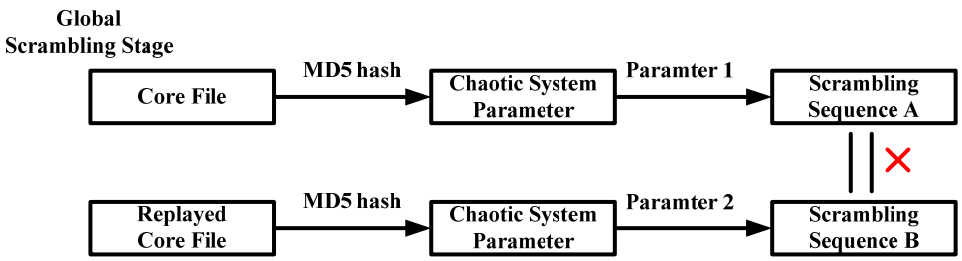


Table 1 Evaluation of the robustness of position permutation

Standard circuit	*LUT size*	*Used resource*	*Blind detection*	$\mu_N(\delta_L)$	$\sigma_N(\delta_L)$	$\rho_N(l, \delta_L)$
SSRAM	128×96	424	Saha	9.34	5.07	0.0832
			Zhang	15.12	7.14	0.0125
			Liang	11.65	5.07	0.0787
			Ours	14.20	7.68	0.0201
DES	256×256	7,064	Saha	23.12	12.51	0.0862
			Zhang	33.75	17.23	0.0156
			Liang	25.55	12.51	0.0742
			Ours	32.12	18.32	0.0325
B22	320×256	7,392	Saha	26.20	12.84	0.0765
			Zhang	37.96	16.14	0.0100
			Liang	28.34	12.84	0.0745
			Ours	34.22	19.45	0.0244
AES	384×320	14,352	Saha	32.25	14.82	0.0937
			Zhang	44.77	18.32	0.0201
			Liang	43.45	14.82	0.0807
			Ours	38.32	22.13	0.0475

4 Experimental results and analysis

In the experiment, we choose the Xilinx ISE tool to design circuit and the experiment platform is Virtex-II XC2V8000. The proposed scheme is compared to algorithms of Saha and Sur-Kolay (2012), Zhang et al. (2013) and Liang et al. (2016a).

4.1 Position scrambling robustness

Position scrambling robustness is an important index to measure the security of blind detection protocols. We use average Manhattan distance $\mu_N(\delta_L)$, Manhattan distance standard deviation $\mu_N(\delta_L)$ and the correlation coefficient $\rho_N(l, \delta_L)$ between original position and Manhattan distance to measure position scrambling robustness. (x_k, y_k) and (x_k', y_k') represent original position coordinate and scrambled position coordinate of the position k. We can use $\delta_{L_k} = |x_k - x_k'| + |y_k - y_k'|$ to indicate the Manhattan distance. The average Manhattan distance is denoted as follows:

$$\mu_N(\delta_L) = \frac{1}{N}\sum_{k=1}^{N}\delta_{L_k} \tag{6}$$

Manhattan distance standard deviation between original position and scrambled position is represented as follows:

$$\sigma_N(\delta_L) = \left[\frac{1}{N}\sum_{k=1}^{N}\left(\delta_{L_k} - \mu_N(\delta_L)\right)^2\right]^{\frac{1}{2}} \tag{7}$$

The correlation coefficient between original position and Manhattan distance is denoted by (8).

$$\rho_N(l, \delta_L) = \left[\rho_N(x, \delta_L)^2 + \rho_N(y, \delta_L)^2\right]^{\frac{1}{2}} \tag{8}$$

$$\begin{aligned}\rho_N(x, \delta_L) &= \frac{\operatorname{cov}(x, \delta_L)}{\sigma_N(x)\sigma_N(\delta_L)} \\ &= \frac{N\sum_{k=1}^{N}x_k\delta_{L_k} - \left(\sum_{k=1}^{N}x_k\right)\left(\sum_{k=1}^{N}\delta_{L_k}\right)}{\left[\left[N\left(\sum_{k=1}^{N}x_k^2\right) - \left(\sum_{k=1}^{N}x_k\right)^2\right]^{\frac{1}{2}}\right.} \\ &\quad \left.\times\left[N\left(\sum_{k=1}^{N}\delta_{L_k}^2\right) - \left(\sum_{k=1}^{N}\delta_{L_k}\right)^2\right]^{\frac{1}{2}}\right]\end{aligned} \tag{9}$$

The calculation of $\rho_N(y, \delta_L)$ is similar to $\rho_N(x, \delta_L)$. It is very clear from the formula (9) that the smaller the correlation coefficient is, the higher the scrambled position robustness. This paper chooses SSRAM, DES, B22 and AES from Xilinx standard cores as the benchmarks in experiments. The position robustness of the blind detection is compared with those of methods in Saha and Sur-Kolay (2012), Zhang et al. (2013) and Liang et al. (2016a). The experimental results are shown in Table 1. Although the proposed scheme adds position fuzzification process in position scrambling algorithm, $\mu_N(\delta_L)$ and $\sigma_N(\delta_L)$ are very similar to other three algorithms and the correlation coefficients of Manhattan distance can be controlled in a very small range. Generally, all the four IP watermarking blind detection algorithms are robust in position scrambling.

4.2 Position scrambling complexity

We choose the swap operation as a measure of computational complexity which is involved in all the blind detection schemes. The experimental platform is Virtex II XC2V8000, and the benchmarks are the SSRAM, DES, B22 and AES. We record average values of four different IP cores at the same stage. The experimental results are shown in Figure 10.

Figure 10 Comparison of the size of core kernel watermarking blind detection (see online version for colours)

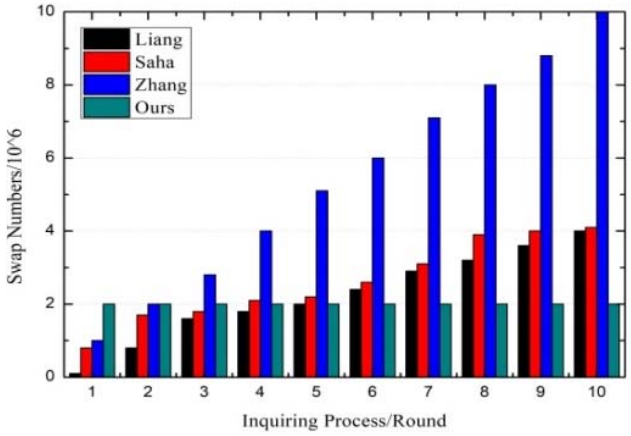


Figure 10 shows the computational complexity in Saha and Sur-Kolay (2012), Zhang et al. (2013) and Liang et al. (2016a) increases linearly with the increase of the inquiry rounds. The proposed scheme just needs one round of inquiring process. Therefore the swap number does not increase with the number of rounds. The computation complexity of the proposed scheme is slightly larger than other schemes during first inquiring. The reason is that scrambling process is divided into global scrambling and position fuzzification process. The swap operations are more than other schemes in single round. But with the increase of the rounds, the advantage of the proposed scheme will be obvious in terms of computational complexity. The other three schemes need to increase the number of rounds to improve the security of protocol. The time costs in a complete blind detection will grow with the increase of rounds. Therefore, the proposed scheme has better performance in real-time detection.

4.3 Watermark capacity

Xilinx Virtex-II FPGA has lots of chip types. IP designs with different complexity occupy different resources of FPGA. Two IP designs need the same number of CLB resources, but the ratio of occupied CLBs to the overall resources is different, as shown in Figure 11.

Figures 11(a) and 11(b) show the resource occupancy of IP core zbt on XC2V500 and XC2V1500. Figures 11(c) and 11(d) are the resource occupancy of wb_cpu01 on XC2V500 and XC2V1000. It demonstrates the same IP core implemented on FPGA with different logic gates having different ratios of resource occupancy. For the same device, the resource occupancy is related to the IP complexity. The greater the IP complexity, the higher the resource occupancy. The relationship between IP core and the chip is shown in Figure 12 which compares the resource occupancy of AES, DES56 and tDES. For the same chip, tDES occupies the most LUTs. As the logic gates in the chip, the resource occupancy decreases.

Figure 11 The CLB occupancy of IP core (see online version for colours)

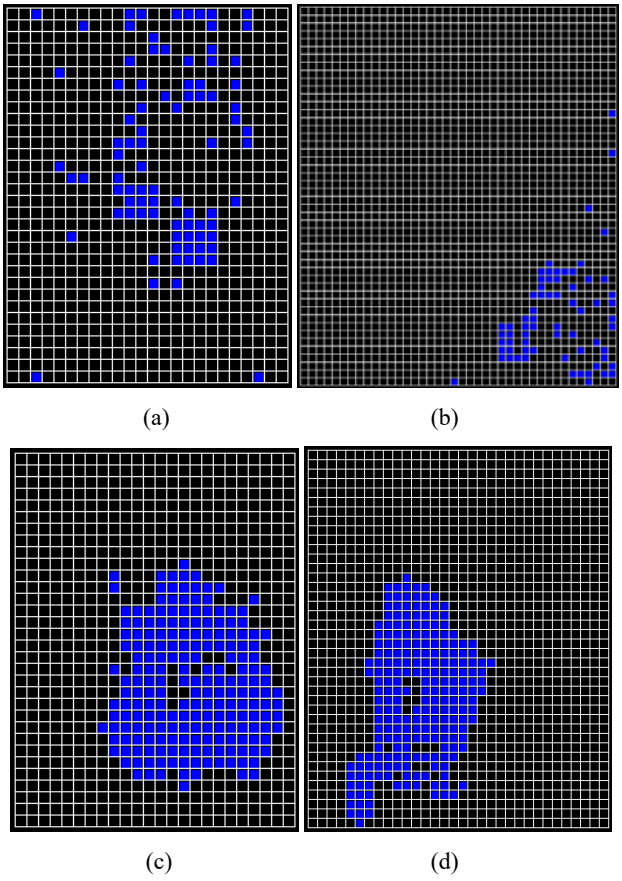
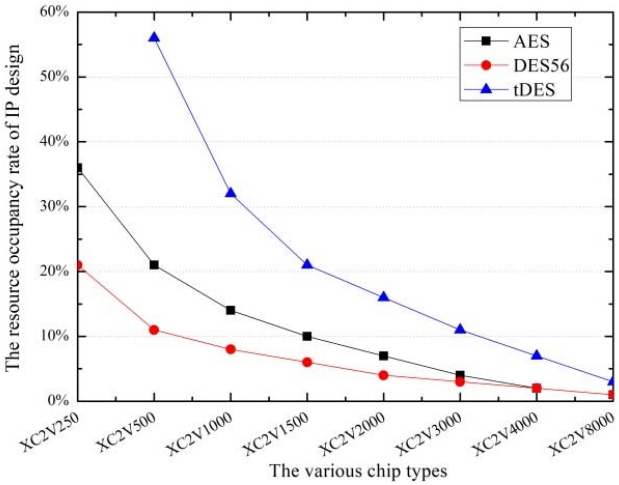
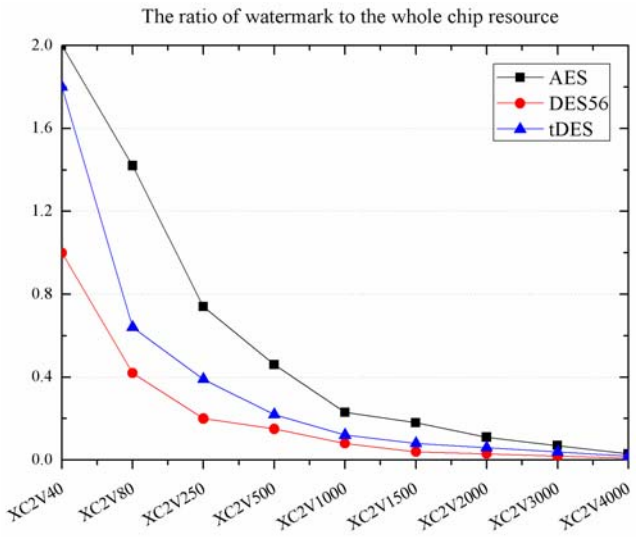


Figure 12 The relationship between IP resource occupancy and the chip type (see online version for colours)



As shown in Figure 13, the proposed watermark embedding algorithm occupies the same size of watermarks in IP designs. But if the IP designs occupy more resources, the resource occupancy rate becomes lower. With advancement in manufacturing technology, Xilinx now produces more chips with higher integration degree.

Figure 13 The resource occupancy of the watermark (see online version for colours)



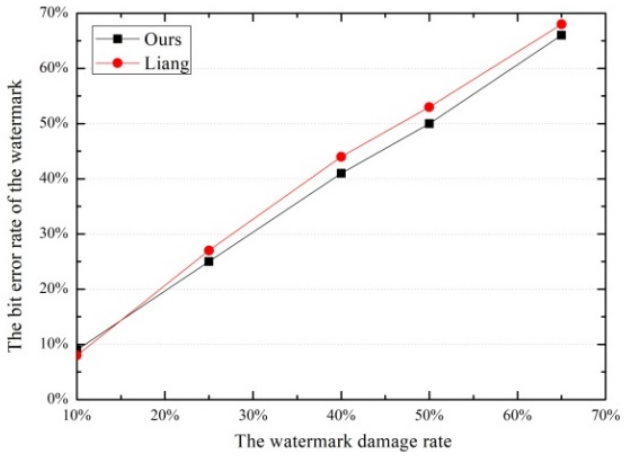
4.4 Robustness

There are two critical metrics to evaluate the robustness of digital watermarks: the normalised correlation (NC) and the bit error ratio (BER). These metrics can be applied to different scenarios. The former is used in watermark detection via correlation calculation and the latter is used in watermark extraction. In the proposed algorithm, the watermark is detection by the extract process. So, BER is used to evaluate the robustness.

$$BER = \frac{100}{B}\sum_{n=0}^{B-1}\begin{cases}1 & w(n) \neq w'(n)\\ 0 & w(n) = w'(n)\end{cases} \tag{10}$$

In (10), B is the length of a watermark. w and w' are the original watermark and the extracted watermark. If the watermark is completely detected, the BER is 0. Otherwise, the BER is 100%. In the proposed algorithm, the watermark length is 128-bit. The robustness evaluation is shown in Figure 14. The x-axis is the watermark damage rate and y-axis is BER. The comparison result shows the BER is in direct proportion to the watermark damage rate.

Figure 14 The BER evaluation (see online version for colours)



4.5 Performance overhead

This section compares the proposed algorithm to the work of Liang et al. (2016a) in resource overhead, as shown in Table 1. The benchmark circuits come from opencores.org. In Table 2, we select six benchmark circuits for evaluation. The FPGAs are the most suitable ones for the benchmark circuits. The original LUTs of the benchmark circuits are listed in the third column. Our proposed algorithm is compared with the method in Liang et al. (2016a).

In Liang et al. (2016a), the zero-knowledge proof requires scrambling positions with Hibert curve. The results show that the proposed algorithm has better performance than that of Liang et al. (2016a).

Table 2 Comparison of LUT resource occupancy

Benchmark circuits	*FPGA*	*#LUT in original IP core*	*Method in Liang et al. (2016a)*		*Ours*	
			#LUT in watermarked IP core	*LUT overhead*	*#LUT in watermarked IP core*	*LUT overhead*
analytic	XC2V250	298	362	21.477%	306	2.685%
des56	XC2V500	692	756	9.249%	700	1.156%
storm	XC2V1500	7,305	7,369	0.876%	7,313	0.110%
aes	XC2V2000	1,378	1,442	4.644%	1,386	0.581%
cpuc	XC2V3000	3,415	3,479	1.874%	3,423	0.234%
rs_dec4	XC2V4000	25,929	25,993	0.247%	25,937	0.031%
Average				5.752%		0.697%

5 Conclusions

With the rapid development of integrated circuit design technology and the popularisation of mobile intelligent devices, integrated circuits play an important role in the field of internet, while IP protection of integrated circuit has raised more concerns. In this paper, we propose a fast blind watermark detection scheme based on position fuzzification. The algorithm introduces the operation of fuzzy position, which breaks the reversibility of the scrambling algorithm. In this case, the number of inquiry rounds is greatly reduced, and the detection efficiency is improved. Experiments show that the proposed scheme greatly reduces computation complexity and has good performance in real-time watermark detection. We will extend our research in multi-dimensional position fuzzification techniques and more suitable methods for calculating the fuzzy radius in order to enhance the stability of the watermark detection algorithm.

Acknowledgements

This research was sponsored by the National Science Foundation of China (Grant 61572188), Xiamen science and technology Foundation (Grant 3502Z20173035), Scientific Research Program of New Century Excellent Talents in Fujian Province University, Fujian Provincial Natural Science Foundation of China (Grant 2018J01570) , the CERNET Innovation Project (Grant NGII20170411).and Natural science fund project in hunan province (2016JJ2058)

References

Bossuet, L. and Torres, L. (2017) *Foundations of Hardware IP Protection*, Springer, Cham.

Castillo, E., Meyer-Baese, U., Garcia, A. et al. (2007) 'IPP@HDL: efficient intellectual property protection scheme for IP cores', *IEEE Transactions on Very Large-Scale Integration Systems*, Vol. 15, No. 5, pp.578–591.

Chang, C.H. and Cui, A. (2010) 'Synthesis-for-testability watermarking for field authentication of VLSI intellectual property', *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 57, No. 7, pp.1618–1630.

Colombier, B., Bossuet, L. and Hly, D. (2016) 'From secured logic to IP protection', *Microprocessors & Microsystems*, Vol. 47, No. A, pp.44–54.

Cui, A. and Chang, C.H. (2012) 'A post-processing scan-chain watermarking scheme for VLSI intellectual property protection', *Circuits and Systems*, pp.412–415, IEEE.

Cui, A., Chang, C.H., Tahar, S. et al. (2011) 'A robust FSM watermarking scheme for IP protection of sequential circuit design', *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 30, No. 5, pp.678–690.

Fan, Y.C. (2008) 'Testing-based watermarking techniques for intellectual-property identification in SOC design', *IEEE Transactions on Instrumentation & Measurement*, Vol. 57, No. 3, pp.467–479.

Fernandes, E., Rahmati, A., Eykholt, K. and Prakash, A. (2017) 'Internet of things security research: a rehash of old ideas or new intellectual challenges?', *IEEE Security & Privacy*, Vol. 15, No. 4, pp.79–84.

Jain, A.K., Yuan, L., Pari, P.R. et al. (2003) 'Zero overhead watermarking technique for FPGA designs', *Proceedings of the ACM Great Lakes Symposium on VLSI 2003*, April, Washington, DC, USA, pp.147–152.

Jung, E., Marchand, C. and Bossuet, L. (2015) 'Identification of embedded control units by state encoding and power consumption analysis', *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, April, pp.523–545, ACM.

Lach, J., Mangione-Smith, W.H. and Potkonjak, M. (2006) 'Fingerprinting techniques for field-programmable gate array intellectual property protection', *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 20, No. 10, pp.1253–1261.

Lach, J., Mangione-Smith, W.H., Potkonjak, M. (1999) 'Robust FPGA intellectual property protection through multiple small watermarks', *Proceedings of the 36th annual ACM/IEEE Design Automation Conference*, pp.849–854, ACM Press, New York.

Liang, W., Huang, Y., Xu, J. et al. (2017) 'A distributed data secure transmission scheme in wireless sensor network', *International Journal of Distributed Sensor*, Vol. 13, No. 4, pp.155–168.

Liang, W., Long, J., Chen, X. et al. (2016a) 'A new publicly verifiable blind detection scheme for intellectual property protection', *International Journal of System Assurance Engineering & Management*, pp.1–13.

Liang, W., Sun, X., Ruan, Z. et al. (2011a) 'A sequential circuit-based IP watermarking algorithm for multiple scan chains in design-for-test', *Radioengineering*, Vol. 20, No. 2, pp.533–539.

Liang, W., Sun, X., Ruan, Z. et al. (2011b) 'The design and FPGA implementation of fsm-based intellectual property watermark algorithm at behavioral level', Vol. 10, No. 4, pp.1597–1601.

Liang, W., Xie, Y., Chen, X. et al. (2016b) 'A two-step MF signal acquisition method for wireless underground sensor networks', *Computer Science and Information Systems*, Vol. 13, No. 2, pp.623–638.

Marchand, C., Bossuet, L. and Jung, E. (2014) 'IP watermark verification based on power consumption analysis', *System-on-Chip Conference*, pp.330–335.

McDonald, J.T., Kim, Y.C., Andel, T.R. et al. (2016) 'Functional polymorphism for intellectual property protection', *IEEE International Symposium on Hardware Oriented Security and Trust (HOST)*, pp.61–66.

Munir, R., Riyanto, B., Sutikno, S. et al. (2007) 'Secure spread spectrum watermarking algorithm based on chaotic map for still images', *Proceedings of the International Conference on Electrical Engineering and Informatics*, Institut Teknologi Bandung, Indonesia, 17–19 June, pp.180–183.

Ngo, X.T., Danger, J., Guilley, S. et al. (2017) 'Cryptographically secure shield for security IPs protection', *IEEE Transactions on Computers*, Vol. 66 , No. 2, pp.354–360.

Qu, G. (2002) 'Publicly detectable watermarking for intellectual property authentication in VLSI design', *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 21, No. 11, pp.1363–1368.

Raju, H., Parthasarathi, D., Saptarshi, N., Samar, S. (2009) 'An internet-based IP protection scheme for circuit designs using linear feedback shift register (LFSR)-based locking', *Proceedings of the 22nd Annual Symposium on Integrated Circuits and System Design: Chip on the Dunes*, pp.665–681, ACM.

Saha, D. and Sur-Kolay, S. (2007) 'Fast robust intellectual property protection for VLSI physical design', *Proceedings of the 10th International Conference on Information Technology*, pp.1–6, IEEE.

Saha, D. and Sur-Kolay, S. (2012) 'Secure public verification of IP marks in FPGA design through a zero-knowledge protocol', *IEEE Transactions on Very Large Scale Integration Systems*, Vol. 20, No. 10, pp.1749–1757.

Schmid, M., Ziener, D. and Teich, J. (2012) 'Netlist-level IP protection by watermarking for LUT-based FPGAs', *International Journal of Computer Applications & Information Technology*, Vol. 1, No. 3, pp.209–216.

Sengupta, A. and Bhadauria, S. (2017) 'Exploring low cost optimal watermark for reusable IP cores during high level synthesis', *IEEE Access*, Vol. 4, No. 1, pp.2198–2215.

Wang, H., He, C., Ding, K. (2004) 'Robust public watermarking based on chaotic map', *Journal of Software*, Vol. 15, No. 8, pp.1245–1251.

Yampolskiy, M, Andel, T.R., McDonald, J.T. et al. (2014) 'Intellectual property protection in additive layer manufacturing: requirements for secure outsourcing', *Proceedings of the 4th Program Protection and Reverse Engineering Workshop*, pp.523–545, ACM.

Zhang, J. and Liu, L. (2017) 'Publicly verifiable watermarking for intellectual property protection in FPGA design', *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 25, No. 4, pp.1520–1527.

Zhang, J., Lin, Y., Lyu, Y. et al. (2013) 'A chaotic-based publicly veriable FPGA IP watermark detection scheme', *Scientia Sinica*, Vol. 43, No. 9, pp.1096–1110.

Zhang, J., Lin, Y., Wu, Q. et al. (2012) 'Watermarking FPGA Bitfile for intellectual property protection', *Radioengineering*, Vol. 21, No. 2, pp.764–771.