A DFA-based Distributed IP Watermarking Method Using Data
Compression Technique

*Jianbo Xu, Jing Long, Wei Liang, Weihong Huang

School of Computer Science and Engineering, Hunan University of Science and Technology,
Hunan 411201, China
E-mail: jbxu@hnust.cn, longjing0404@163.com, idlink@163.com, whhuang@hnust.cn
doi : 10.4156/jcit.vol6.issue8.18

Abstract

The IP (Intellectual Property) protection has been widely concerned by more semiconductor
companies and research institutions, due to the rapid advances in deep sun-micron integrated circuits.
A DFA (Deterministic Finite Automaton)-based distributed IP watermarking method is presented by
using data compression technology. DFA is used for generating pseudorandom key stream and
encrypting the original signature. The cipher text is generated, and then transformed into embedding
coordinates and watermarks by specific data compression technology. Finally, the watermarks are
denoted by some identical logic modules, and embedded in the LUTs (Look-up Table) of FPGA (Field
Programmable Gate Array) design. The contribution lies in the identification of the same ownership
with much less watermark bits than most of other methods. The experiment results on Xilinx Virtex II
Pro XC2VP4 FPGA show high security, a negligible resource increase and low circuit delay.

Keywords: IP Protection, DFA, IP watermarking, LUT, FPGA

1. Introduction

The emergence of reusable IP modules in IC systems drastically reduces development cycle and
design cost[1]. Reuse-based design has become a mainstream method, while it brings about new risks
and challenges simultaneously. Most of IP designs need much time and effort. On one hand, IP vendors
always expect their IP designs will not be misappropriated and the identification is easy to perform. On
the other hand, IP users also hope the reusable IP cores they buy are credible, thus the design risk will
be greatly reduced[2]. Generally, the reusable IP cores are propagated in netlist file or other forms. In
this case, they are easily copied and propagated by unauthorized users, the security is under threat.
Therefore, legal propagation of reusable IP designs relies naturally on IP protection methods.
Traditional IP protection methods primarily depend on law and own protection[3]. Here, the former
uses the means, such as patent, ownership and contract to deter the illegal users, thus protect the
intellectual property, while encryption, watermark identification[4-5] and license et al. are used as own
protection methods, in which watermark identification is widely concerned. In this method, IP designs
act as the carrier, specific digital information is hidden in the circuits for ownership identification and
tracing infringement[6].

In recent years, FPGA becomes popular for IP design in IC system, therefore, it is critical to protect
the FPGA-based IP designs. Since FPGA-based IP watermarking was first proposed by John Lach, et
al.[7-8], many researchers have been developing FPGA-based IP watermarking successively. Adarsh
Kumar Jain et al.[9] proposed a zero overhead watermarking method by modifying the path delay.
Encarnación Castillo et al.[10] proposed to embed watermark at the behavioral design level, adding
the information into the locations between used LUT and unused LUT of FPGA design. The hardware
overhead is caused by the watermark extraction. If the watermark extraction circuit detects the specific
input sequence, it will route the watermark to the output. Wei Liang et al.[11-12] have investigated
watermarking methods at physical design level and behavioral design level. A chaotic IP watermarking
at physical layout level based on FPGA has been developed. The alternating structure of two 1D
chaotic maps is used for better security in the method. The performance degradation in terms of circuit
area, speed and power overhead is negligible and the method offers high robustness. Moritz Schmid et
al.[13] proposed to restrict the dynamically addressable part of the logic table, thus freeing space for
insertion of signature bits into LUTs. Simultaneously, the functional LUTs are converted to LUT-based RAM or shift registers for preventing deletion due to optimization. When infringement occurs, IP identification is performed by extracting watermark from the bitfile of the FPGA design. Debasri Saha et al. proposed a method for embedding watermark at FPGA designs and developed a Zero-Knowledge Protocol for watermark verification[14].

A number of IP protection methods have been developed recently. However, most of the methods have poor security and the watermark embedding may lead to many additional constraints. These constraints generally result in increase of hardware resource, making the performance degradation. A DFA-based distributed IP watermarking method is proposed for solving the problems such as poor security and high resource overhead. The contributions of the method lie in two aspects as follows:

(1) A safe distributed watermarking method is proposed. In the method, identical logic configurations are integrated with the original design, making the watermark randomly distribute in the design. In this case, it is hard for unauthorized user to detect the watermark positions. As for the VLSI (Very Large Scale Integrated Circuits), the difficulty for attackers to remove the watermark is equal to redesign.

(2) An effective data compression mechanism is developed. The ownership signature is encrypted and then transformed using the proposed data compression method. Finally, the actual watermark which is embedded in the design has been reduced drastically. Therefore, the performance impact on hardware overhead is negligible.

The rest of the paper is organized as follows: Section 2 discusses DFA(Deterministic Finite Automaton) encryption method and the preprocess of signature cipher text using data compression mechanism; describes the processes of watermark embedding and extraction; Section 3 presents experimental results and analyzes the performance of the proposed method and the paper is summarized in Section 4.

2. DFA-based distributed watermarking method

The proposed method consists of watermark generation, watermark embedding and extraction. A compression mechanism is proposed for enhancing the security and reducing the watermark. The signature is encrypted by DFA encryption method and then transformed into watermark coordinates and watermark bits using the compression mechanism. Finally, the watermark is denoted by the identical logic and embedded in the design for ownership verification.

2.1. Watermark generation

To improve the security, DFA theory is introduced for signature encryption. This section gives the principle of DFA encryption and then introduces the watermark generation using compression mechanism.

2.1.1. DFA Encryption

Deterministic Finite Automaton is a mathematical model with disperse input set and output set[15]. Given DFA $M = \{Q, \Sigma_1, \Sigma_2, \phi, \psi, q_0, Z\}$, where state set $Q = \{q_i | i = 0, 1, 2, ..., l\}$. Assume that $\Sigma_1 = \{\lambda_i | i = 1, 2, ..., m\}$ is a set of input symbols. $\Sigma_2 = \{k_i | i = 1, 2, ..., n\}$ is a set of output symbols. Transition function $\phi$ is the map from $Q \times \Sigma_1$ to $Q$. Output function $\psi$ is the map from $Q \times \Sigma_1$ to $\Sigma_2$. $q_0$ denotes initial state and $Z$ represents termination state. The symbol set recognized by $M$ is denoted by $L(M)$. Therefore, $M$ is used as a key generation for generating key stream $k = k_1k_2...$ according to the following steps.

Randomly choose a sequence $\lambda = \lambda_0\lambda_2\ldots\lambda_n \in L(M)$ from $\Sigma_1$ as initial key $K$. For each $j > 0$, set $u_j = \lambda_0 (mod r)$, $\phi(q_0, u_j) = q_1', u_j = q_1$, $\phi(q_{j-1}', u_j) = q_j', u_j = q_j$, $\phi(q_j', u_j) = q_{j+1}'$, $u_j = q_{j+1}$. Then we have $k_1 = \psi(q_0, u_j)$, $k_2 = \psi(q_1', u_j)$, ..., $k_j = \psi(q_j', u_j)$, ... . In this case, the initial key $K$ is transformed into sequence
A DFA-based Distributed IP Watermarking Method Using Data Compression Technique
Jianbo Xu, Jing Long, Wei Liang, Weihong Huang
Journal of Convergence Information Technology, Volume 6, Number 8, August 2011

$k_1, k_2, \ldots \in \Sigma_j$ by using maps $\varphi$ and $\psi$, that is the key stream generated by $M$ using initial key $K$. The Figure 1 shows the workflow of DFA stream cipher. The initial key $K$ is transformed into key stream $k_1, k_2, \ldots$. Meanwhile, the plain text is divided into groups $p = p_1, p_2, \ldots$. We encrypt the $i$th element $p_i$ of the plain text by using the $i$th element $k_i$ in key stream $k$. The encryption function is $E = E_k(p_i)$. $D = D_k(c_i)$ is the corresponding decryption function, satisfying $D_k(E_k(p_i)) = p_i$.

**Figure 1.** The workflow of DFA stream cipher

### 2.1.2. Watermark Preprocess

This section introduces the watermark transformation using compression mechanism. The cipher text of signature is transformed into watermark coordinates and watermark bits. The concrete implementation is shown as follows:

- Step 1: Encode ownership information such as company name, signature and trademark, generate the plain text stream $P = \{p_i \mid i = 0,1,2,\ldots\}$
- Step 2: Select a initial key $K$ and use DFA $M$ for generating key stream $k = k_1, k_2, \ldots$. After that we encrypt the plain text $P$ with encipher $E$, generating cipher text $C$.
- Step 3: Divide cipher text $C$ into groups $Grp = \{grp_i \mid i = 0,1,2,\ldots\}$, where the length of each group is 8 bits, denoted by $\|grp_i\|$. Then we transform the group $grp_i$ into decimal number $Dec = \{dec_i \mid i = 0,1,2,\ldots\}$
- Step 4: Factorize each element $dec_i$ in $Dec$ with specific rules for watermark compression and we get $X \times Y + B$, here $X$ and $Y$ represent the horizontal coordinate and vertical coordinate respectively, that is $Pos = \{(X_i, Y_i) \mid i = 0,1,2,\ldots\}$. $B \in \{0,1\}$ denotes the watermark bit $Wm = \{wm_i \mid i = 0,1,2,\ldots\}$ which is embedded in the position $(X,Y)$. The Factorization makes the signature denote by the combination of coordinates and embedded bits.

Supposing that the ownership signature is “JINGLONGHNUST…”, we perform Step 1 to Step 3 and get the cipher text “88123376BA90B1F3…”. By using Step 4 and 5, the cipher text is then transformed into the watermark coordinates and watermark bits, as shown in Table 1. Assume that the length of encrypted signature is 128 bits, the compression mechanism, that is factorization, makes the watermark drastically reduce to 16 bits, which steeply decreases the resource overhead.
2.2. Watermark Embedding

FPGA is a large scale programmable logic device, which consists of configurable logic block (CLB), input/output block (IOB) and interconnect resource (IR). Generally, CLB is composed by flip-flops (FF) and lookup table (LUT). Most of FPGA-based designs have a number of unused resources, which offer an new approach for FPGA-based watermarking. The proposed method adds some redundant identical logics into original design using FPGA structure, which has no impact on the normal function. Firstly, depending on the thought of identical function and analyzing the circuit features, we design the identical logic for indicating “0” and “1”. Then the compression mechanism is used for the signature transformation. Finally, the embedding process is performed for watermark insertion. The proposed method has low hardware overhead due to the reduction of watermark bits by using the compression.

![Functional Signal](image)

![Identical Logic](image)

(a) Design thought of identical logic  
(b) Structure of identical logic “0” and “1”

**Figure 2.** The design and implementation of identical logics

Figure 2(a) shows the design thought of identical logic. When the states of two flip-flops are opposite, output of the AND gate will always be zero. After the functional signal and the AND output through the OR gate, the value of functional signal will not change. In this case, whatever the functional signal is, the OR output will be the same with the input. It is demonstrated that the addition of the redundant logic has no interference with the normal function. We design the circuits for indicating logic “0” and logic “1” on the basis of identical function, as shown in Figure 2(b). The implementation of watermark embedding is shown in Figure 3. $I_1$ and $I_2$ represent the inputs of functional logic $f(x_1, \ldots, x_n)$ in FPGA design, $O$ denotes the output. If the watermark is “0”, we could add the redundant identical logic “0” after functional logic $f(x_1, \ldots, x_n)$. Finally, route the signal $O_{0}$ and $O$ to OR gate, the output after embedding watermark is indicated by $O'$. Actually, the values of signal $O$ and $O'$ are identical.
A DFA-based Distributed IP Watermarking Method Using Data Compression Technique
Jianbo Xu, Jing Long, Wei Liang, Weihong Huang
Journal of Convergence Information Technology, Volume6, Number 8, August 2011

In the DFA-based distributed IP watermarking method, watermark embedding is performed by the following steps.

1. Follow the implementation flow of IP design and build the original IP with no watermark by using hardware description language.

2. Configure the FPGA device in the integrated development environment ISE; perform compiling, synthesis, simulation, and finally generate the physical layout; According to the actual requirement, program relevant additional conditions in the user constraints file for restriction.

3. In section 2.1, the watermark positions \(\{(X_i, Y_j) | i, j = 0, 1, 2, \ldots\}\) are generated; With the positions, we could search the corresponding LUT resource for watermark embedding.

4. According to the watermark bits generated in section 2.1, we configure corresponding identical logics in the positions. In this way, the watermarks are embedded in the original design. Once the infringement occurs, the IP owner could apply to the third party for ownership verification.

2.3. Watermark Extraction and Identification

When the IP design is suspected to be misappropriated, the owner could apply to the neutral third party for ownership verification. If the ownership information is extracted in the suspect design, the identification is proved. The watermark extraction steps are shown as follows.

1. For the IP owner, he will reserve the watermark positions \(\text{Pos}\) for watermark verification. Using the watermark positions, all of the watermark LUTs are easily found.

2. After locating the watermark positions, we extract the identical logics sequentially and record the specific indication of each identical logic.

3. By the inverse process of watermark transformation, we transform the extracted logics into “0” or “1” bit as remainder. Multiply the corresponding horizontal coordinate \(X\) and vertical coordinate \(Y\), simultaneously, add the product with the remainder, the sum is the decimal number corresponding to each group of cipher text.

4. Transform the decimal number into hexadecimal number, and recombine sequentially to get the cipher text \(C\).

5. With the reserved key stream \(k\) generated by DFA key generator, use the decipher \(D\) for decrypting \(C\) and get the plain text \(P\). \(P\) is the signature after encoding. Here, we decode the sequence \(P\) and the original signature is generated. If the extracted signature is consistent with the original one, the IP ownership is demonstrated.

3. Experiment Result and Analysis

In this section, a series of experiments have been conducted to evaluate the DFT-based distributed watermarking method. In order to verify the properties of the proposed method, we applied our method to several encryption text circuits[16] on the Virtex II Pro XC2VP4 FPGA platform. We will analyze the performances in terms of resistance to attacks, security, resource overhead and delay time.
3.1. Resistance to Attack

Resistance to attacks is the ability of watermark being extracted correctly after various unauthorized attacks. General attack methods include invasion attack, removal attack and power analysis attack et al. For invasion attack, the packages of chip or circuit are removed. The attackers get information by observing, control and intervene the internal chip and the communication between chip and circuit. It is more difficult than other attacks due to the requirement of expensive equipments. In removal attack, the attackers attempt to remove the watermark by various means. For the proposed method, the watermarks are distributed randomly in the design and the watermark positions are relatively fewer. If the attackers perform removal attacks on large scale design, the effort is almost identical with redesign. Power analysis attack is a threat to the safe FPGA-based designs. The attacks are performed by analyzing the power curve to get relevant sensitive information, according to which the attack could destroy or remove the watermark[17].

In the proposed method, the signature has been transformed using compression mechanism. The watermark are dispersed in all of the design, which may effectively improve resistance to attack. For verification of this property, power simulation tool is used to get the approximately actual power information. Then we perform power analysis attack on the power information for property verification. If the attack is successful, the differential value of the result is greater than 0 and closer to 1. The closer to 1, the weaker the resistance to attack is. On contrary, attack failure makes the differentia value close to 0. The closer to 0, the stronger. In this section, the proposed is compared with the methods in [13] and [14]. We perform 40 times power analysis attacks and increase the sample number by 50 each time. All the sample for power analysis attacks are 2000. The experimental results in Figure 4 show that in the proposed method, the differential value is closer to 0 by comparing with [13] and [14]. Therefore, the stronger resistance to attack is demonstrated.

![Figure 4. Experiment result of power analysis attack](image)

3.2. Coincidence of probability

In the verification model of IP watermarking, the coincidence of probability $P_c$ is used for security evaluation of distributed watermarking method. $P_c$ is the probability for a non-watermark design carrying the watermark by coincidence. In accordance with the design thought of the proposed method, the watermark is generated by transforming the signature with the compression mechanism. Assume that the length of watermark bits is $m$ and the number of “0” in watermark is denoted by $m_0$. The number of resources in the design is represented by $n$. Then there are $C_n^m$ situations to select $m$ watermark positions satisfying the constraints from $n$ resource positions in the design. Supposing that the probability for the selected positions just containing the logic “0” or “1” is $p_0$ or $p_1$ respectively. Therefore, the coincidence of probability $P_c$ could be computed as follows:
A DFA-based Distributed IP Watermarking Method Using Data Compression Technique
Jianbo Xu, Jing Long, Wei Liang, Weihong Huang
Journal of Convergence Information Technology, Volume 6, Number 8, August 2011

\[ P_e = \frac{1}{C_m} (p_e)^{(m-m_0)} \]

In general, \( P_e \) is expected to tend to 0 for better security. To evaluate security of the proposed method, five encryption cores are used for \( P_e \) evaluation. We test the \( P_e \) rate of change in the case of embedding different number of watermark, the results are shown in Figure 5. It shows that, as for different IP design, \( P_e \) tends to 0 with the increase of watermark. Security of the proposed method is guaranteed.

![Figure 5. \( P_e \) rate of change in various circuits](image)

### 3.3. Overhead

The additional hardware overhead of watermarking circuit is the overhead increase after embedding watermark, such as resource overhead and delay time. In this section, five encryption designs are used as test circuit. The experiments are conducted on Xilinx Virtex II Pro XC2VP4 FPGA platform for overhead evaluation. The experimental results are shown in Table 2, here columns “Cells” and “Timing” are respectively the number of slices occupied by original design and the delay time of critical path, “\( \Delta S(\%) \)” and “\( \Delta T(\%) \)” respectively represent the resource change rate and delay time change rate after embedding watermark.

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Original design</th>
<th>64 bit watermark</th>
<th>128 bit watermark</th>
<th>512 bit watermark</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cells(slices)</td>
<td>Timing(ns)</td>
<td>( \Delta S(%) )</td>
<td>( \Delta T(%) )</td>
</tr>
<tr>
<td>DES56</td>
<td>317</td>
<td>4.679</td>
<td>1.262</td>
<td>0.173</td>
</tr>
<tr>
<td>RSA</td>
<td>490</td>
<td>9.130</td>
<td>0.412</td>
<td>0.129</td>
</tr>
<tr>
<td>AES</td>
<td>646</td>
<td>4.532</td>
<td>0.464</td>
<td>-0.091</td>
</tr>
<tr>
<td>MD5</td>
<td>1171</td>
<td>18.018</td>
<td>0.341</td>
<td>0.054</td>
</tr>
<tr>
<td>SHA</td>
<td>1695</td>
<td>12.765</td>
<td>0.236</td>
<td>-0.078</td>
</tr>
</tbody>
</table>

In Table 2, DES56 circuit occupies the fewest resources, while SHA circuit occupies the most. By embedding incremental watermark, the occupied resources tend to increase but the rate of change is not obvious. It results from the watermark embedding method by using identical logics configuration. Simultaneously, it can be seen that the delay time change rate are negligible. As the design of identical logic adopts a few basic logic gates and the watermark positions are of small number, the delay time and resource overhead caused by watermark embedding are acceptable.

Figure 6 compares the proposed method with methods in [13] and [14] in terms of resource and delay time. The RSA circuit is used for property verification. As seen in Figure 6(a), with increasing watermark, the proposed method has certain superiority in resource occupation by comparing. Although the resource growth tends to ascend, the curve becomes gentle after embedding more watermarks. The growth of delay time in three methods shows an overall rising trend. In contrast, the delay time in the proposed has lower rate of change, as shown in Figure 6(b). Consequently, the proposed method has certain advantage on overhead and timing.
4. Conclusion

A new DFA-based distributed IP watermarking method using compression technique is presented. It provides an effective way for solving problems in the existing watermarking method. The ownership signature is encrypted with DFA principle. The cipher text is then transformed into watermark positions and watermark bits by using the compression mechanism. The watermark bits are embedded into the corresponding positions in the circuit through identical logic configuration. The experimental results show no impact on logic function. By comparing with other methods, the proposed method will cause a slight increase of resource and delay time, while the properties in terms of security and resistance to attack are better. In future, we will study on effective watermarking methods at various design levels, which include watermark optimization at high design level, insertion of test circuit and the implementation of watermark embedding circuit.

5. Acknowledgments

This work is supported by the Scientific Research Fund of Hunan Provincial Education Department (Grant Nos. 09A027, 09C403), Natural Science Foundation of Xiangtan United Fund of Hunan Province (Grant No. 09JJ9006) and Innovation Fund Project for Graduate Student of Hunan University of Science and Technology(S100119).

6. References

A DFA-based Distributed IP Watermarking Method Using Data Compression Technique
Jianbo Xu, Jing Long, Wei Liang, Weihong Huang
Journal of Convergence Information Technology, Volume 6, Number 8, August 2011


