A New Methodology to Protect PCBs from Non-Destructive Reverse Engineering

Zimu Guo, Bicky Shakya, Haoting Shen, Swarup Bhunia, Navid Asadizanjani, Mark Tehranipoor, and Domenic Forte

Electrical and Computer Engineering Department, University of Florida

Abstract: *Reverse engineering of electronic hardware has been performed for decades for two broad purposes: (1) honest and legal means for failure analysis and trust verification; and (2) dishonest and illegal means of cloning, counterfeiting, and development of attacks on hardware to gain competitive edge in a market. Destructive methods have been typically considered most effective to reverse engineer Printed Circuit Boards (PCBs) – a platform used in nearly all electronic systems to mechanically support and electrically connect all hardware components. However, the advent of advanced characterization and imaging tools such as X-ray tomography has shifted the reverse engineering of electronics toward non-destructive methods. These methods considerably lower the associated time and cost to reverse engineer a complex multi-layer PCB. In this paper, we introduce a new anti–reverse engineering method to protect PCBs from non-destructive reverse engineering. We add high-Z materials inside PCBs and develop advanced layout algorithms, which create inevitable imaging artifacts during tomography, thereby making it practically infeasible for an adversary to extract correct design information with X-ray tomography.*

Keywords: Reverse Engineering, Printed Circuit Boards (PCBs), X-ray tomography, PCB Trust.

Introduction

Reverse engineering (RE) is a process where the goal is to reproduce, duplicate, or enhance a design based on the study of an original object/system. For electronic systems, reverse engineering could be performed at multiple levels of design abstraction: (1) microchip, (2) printed circuit board (PCB), and (3) system levels. PCBs are used in nearly all electronic systems to provide mechanical support and electrical connection of all electronic components including microchips, passive components (e.g. resistors, capacitors, etc.) and sensors. With increasing complexity of electronic hardware, PCBs contain significant design information for a system, and hence RE of PCBs has emerged as an essential tool for design debug and repair through failure analysis and fault isolation. On the other hand, RE has also been exploited by attackers for malicious intents, such as to steal design information, clone or tamper a PCB design. Since modern PCBs typically consist of multiple layers, reverse engineering involves obtaining the internal structure and connections of all layers through either a destructive or non-destructive process.

In a destructive process, delayering is followed with imaging of every layer before the next round of material removal. On the other hand, a non-destructive process consists of imaging tomography, which can be used to image the whole system or board without delayering. In either case, the analysis can be automated or manual, and ultimately results in a netlist [1]–[3] that can be used to reproduce the system. The industry has shown an interest in shifting to reverse engineering practices based on non-destructive methods due to their lower costs, the shorter time period that it takes to reverse engineer a PCB, and potential application to manufacturing test for detecting faults or trust issues. Their non-destructive nature also allows more margin for error during the procedure and the PCB can be used for other purposes afterwards [4]. The authors have introduced a complete framework for PCB reverse engineering using X-ray tomography in prior work. As illustrated in Figure 1,

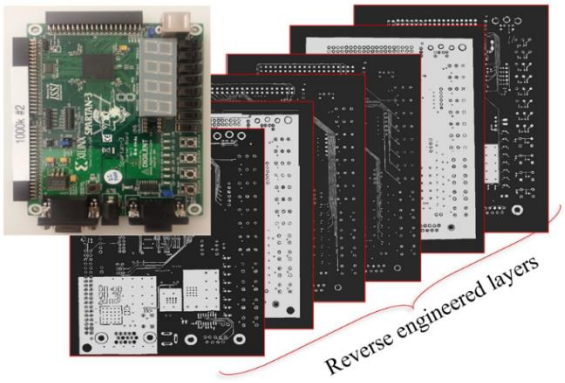


Figure 1. Reverse-engineered Xilinx Spartan 6 PCB.

all the connection information including metal traces and vias have been very well extracted based on the proposed imaging and image-processing algorithms [2], [11]-[13]. The images are representing the information on all six layers of a commercial Xilinx Spartan 6 PCB. Although this technique is applied on one commercial PCB as a case study, the procedure is described in a general way making it applicable to PCBs with various number of layers and other structural differences. Regular X-ray tomography systems can detect features of dimension as small as one micron. This would be the only limitation for this method, though most of the commercially available PCBs in the market currently have much larger features.

In this paper, we introduce a new methodology to protect PCBs against non-destructive attempts for reverse engineering. It aims at protecting PCBs against RE for malicious purposes – in particular cloning and tampering. Our methods are based on incorporation of high-Z material inside PCB layers that can create strong noise and artifacts in the reconstructed images in a destructive way, where the features can no longer be extracted after reconstructing the 3D image. Although this might appear very simple to implement at first glance, one has to consider that there are not many options for materials that can be used in the PCBs with minimum modification to the PCB structure and can still create enough noise for protection against non-destructive RE. In addition, the location of the high-Z material in PCB layers is also an important consideration and should be optimized to achieve high enough noise to prevent reverse engineering without incurring unacceptable cost.

X-Ray Tomography and Image Reconstruction Principles

The basic operation of 3D X-ray tomography is acquiring two-dimensional (2D) images while the sample rotates. The 2D projections are collected from many different angles depending on the feature size and the required image quality. The object properties, such as its dimension and material density, are important to consider in selection of the tomography process parameters, which include the following.

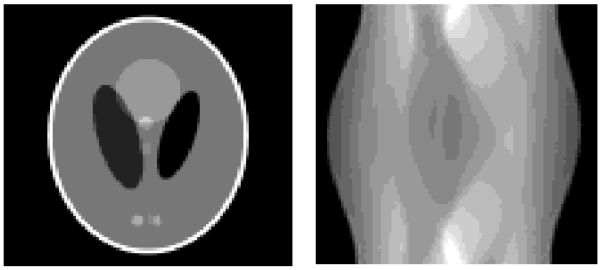
- *Source power*: correlated to the X-ray energy and amount of penetration
- *Detector objective*: determines the field of view and resolution range
- *Filtering*: controls the dose which allows higher energy X-rays to penetrate
- *Distance of source and detector from sample*: inversely proportional to number of transmitted X-rays
- *Number of X-ray projections*: identifies the angular steps as sample rotates
- *Exposure time*: linearly related to counts and determines the total time and, consequently, the cost of scanning

These parameters can affect the pixel size and signal-to-noise ratio (SNR), which must be optimized based on the region of interest.

If the original sample is a real space object, *f(x,y)*, the 2D projection is in fact equal to the Radon transform of *f(x,y)*, which is the line integral through *f* and along one ray line from the X-ray source to the detector

$$Rf = \int_L f(x,y)ds \qquad \text{Eqn.1}$$

where, *Rf* is representing the Radon transform of *f*. On the other hand, according to mathematical principles, a discrete sampling of the object and the object's Radon transform are geometrically equivalent to the sampling of the same object by a transmitted signal, which is the 2D projection in X-ray tomography. This geometric equivalency is the basis for most of the 3D reconstruction algorithms used in different tomography systems, such as X-rays, focused ion beam (FIB), and transmission electron microscopy (TEM). This translates into the fact that one can reconstruct an object by implementation of the "inverse Radon transform" on the acquired 2D projections. In Figure 2, the original image of an object and its Radon transform are presented.



(a) (b)

Figure 2. a) Original image and b) its Radon transform of an object [10].

Radon transform is the basis for other mathematical tools, such as center slice theory (CST) and back projection (BP) theory, used to better reconstruct the 3D images [10]. Such techniques have been used

widely in medical imaging, industrial imaging, acoustic imaging, and more.

Imaging Interference in Tomography

For an object to be visible in an X-ray image, it should have enough physical contrast with the other surrounding material. This contrast is in direct relation with the associated material of the object, its **physical density**, and its chemical composition or **atomic number**. These two parameters (atomic number and physical density) are the two most important parameters affecting X-ray attenuation coefficient. When an object is physically different, it absorbs either more or less X-radiation than the surrounding objects and creates a shadow on the X-ray detector. Higher density and atomic number will result in higher attenuation and darker shadow on the detector. The third factor that affects the contrast is **thickness** in the direction of the X-ray beam. For example, a thick material with lower density will have almost the same contrast in an X-ray projection as a thin material with higher density.

It is very important to mention that whenever a dense material with a higher atomic number is located next to a light material with a lower atomic number, the 2D projection will mostly represent the heavy material. In such cases the 2D projection image from certain angles, where the X-ray cannot pass through one material at a time and, therefore, will mostly present the heavier material. This can cause serious noise in the final 3D image, depending on the total number of such projections. If the light material in the sample structure is blocked from being directly exposed to X-radiation in many angles, it is very difficult to have a clear image of its structure after final reconstruction.

As shown in Figure 3, proper high-Z materials can be used in PCBs to block the X-ray transmission through PCB layers, as shown in Figure 3. Without sufficient transmission X-ray, the 2D imaging and 3D tomography will be difficult. The blocking material can be prepared as thin sheets for PCBs. During the bounding of PCB layers, the blocking material is taken care of as core layers, placed between prepregs. Or, the blocking material can also be prepared as particles, mixed into the resin-based prepreg for the bounding of PCB layers. By choosing proper particle diameters and concentrations, sufficient effective thickness of blocking materials can be obtained as well. The flexible choice of forms can significantly lower the material casting cost, especially for hard and fragile materials like tungsten. As most candidate materials are metals with high conductivity, traditional PCB processing should be modified to ensure a good isolation between the blocking material and circuits. Steps similar to the surface isolation might be necessary for certain circuit layers. In addition, to avoid circuit-shortage between the blocking materials and the vias in PCBs, both the pattern of blocking materials and the location of vias should be considered during the PCB layout design. For instance, for sheet-blocking materials, a hole with a diameter safely larger than the vias should be reserved where a via may go through. For powder-blocking materials that are mixed with epoxy, the powders should be defined within certain regions so that the vias can circumvent the powders.

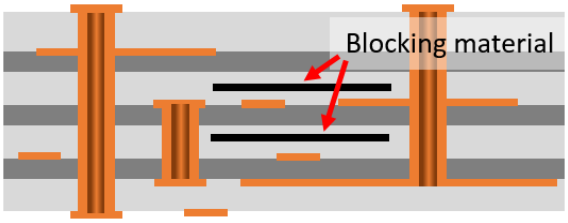


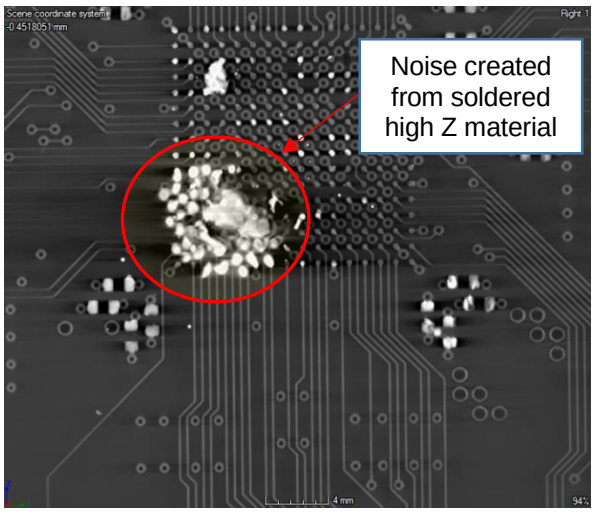
Figure 3. PCB with implemented blocking materials.

These can be used as a basis for anti–reverse engineering of non-destructive methods that use X-ray tomography. PCBs are mostly made of thin and low-Z material including copper, aluminum, etc. Although it is easy to run X-ray tomography on such materials, high-Z material presence in the region can create noise and/or block transmission X-ray by absorption as described above.

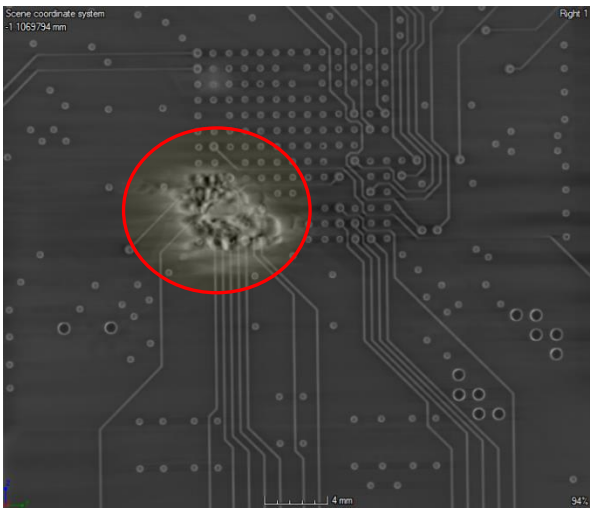
Experimental Results

There are different candidates for high-Z materials that can be used in PCBs—materials such as tantalum, gold, lead, etc. The atomic number for these high-Z materials is in the range of 70–80, which is much higher than the atomic number of copper (29) or aluminum (13). The high-Z materials can cause major shadowing and noise on the results. Although gold and lead are good candidates in terms of attenuation, their price and safety are limiting factors, so tantalum and tungsten are the most promising candidates for the purpose of this paper.

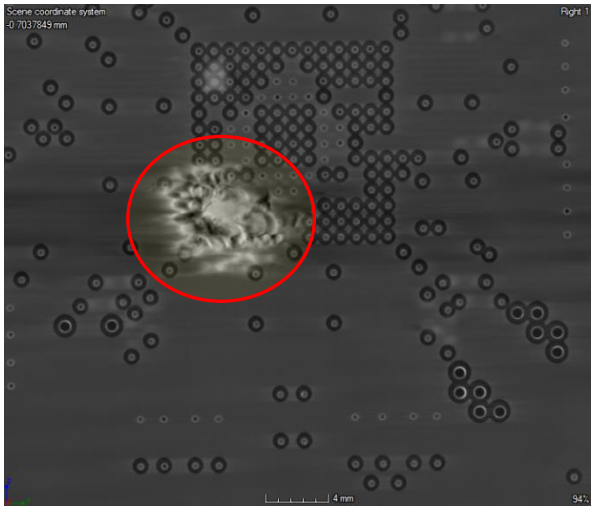
In this paper, we have used soldering material, which has an atomic number close to tantalum. As seen in Figure 4 from a successful X-ray tomography image, one can easily detect all traces and connection in all



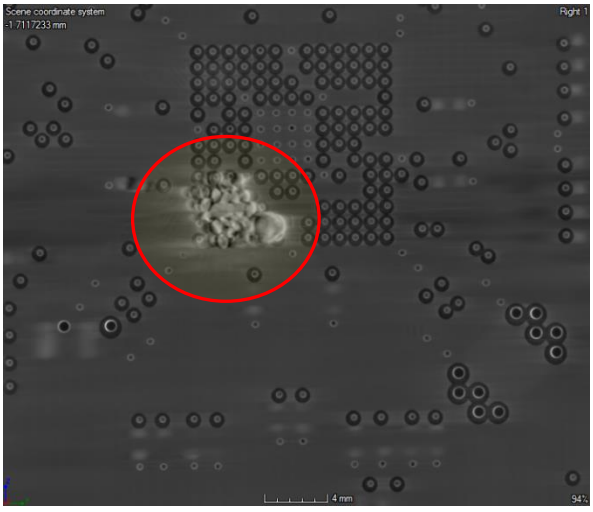
Layer 1



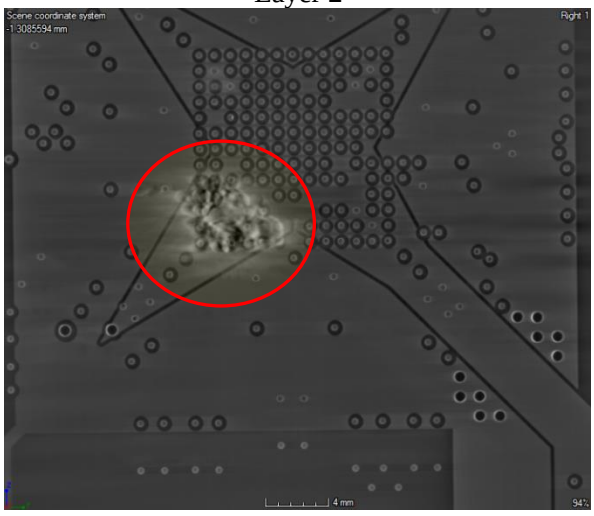
Layer 4



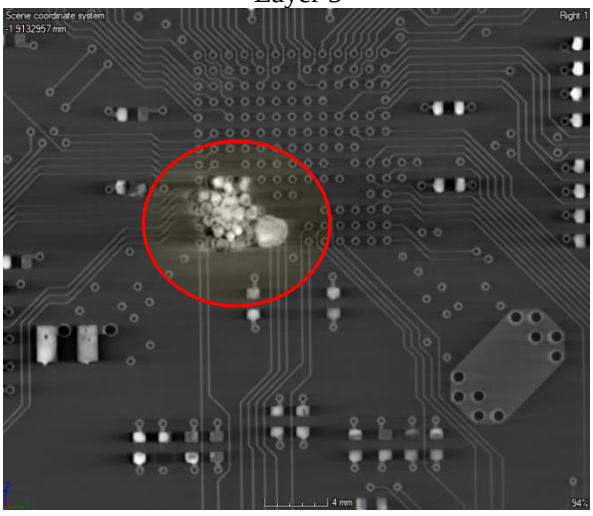
Layer 2



Layer 5



Layer 3



Layer 6

Figure 4. X-ray tomography images of PCB. Traces and connections are not detectable in the noisy region due to the artifacts from extra material.

layers expect where the high Z material has created noise. These dense materials will absorb most of the X-rays in that region and will act as a shield. As the sample rotates during tomography, some features may come out of the coverage of these high-Z material, but there will be only a few angles from which the X-ray can penetrate. As a rule of thumb for tomography of a flat sample, one needs to take proper 2D projections from at least 0 to 180 degrees while the sample rotates. Due to the complexity and small size of features in modern PCBs, it is very easy to cover a critical region with high-Z material and prevent using X-ray tomography for reverse-engineering purposes.

X-Ray Absorption Calculation

To estimate the X-ray absorption (A) on the unit area in the materials, we need to know the intensity of incident X-ray (I_0) and the reflected/refracted X-ray (I_{rfl}/I_{rfr}) on the sample surface. Based on the X-ray emission power, the emission angle and the distance between X-ray source and the tested sample, I_0 can be directly obtained. Since the decrement (δ) of X-ray for most materials is very small (less than 10^{-6} by several orders of magnitudes) [9], the index of refraction (n = 1- δ) is very close to the unit and thus the reflection of X-ray from the sample surface can be neglected,

regardless of the incident angle (i.e., $I_{rfl} = 0$, $I_{rfr} = I_0$). The X-ray intensity (I) in materials is given by [6]:

$$I_t = I_0 \cdot \exp(-\frac{\mu}{\rho} \cdot \rho \cdot t) \qquad \text{Eqn. 2}$$

where I_t is the X-ray intensity in materials at depth of t from the surface, μ/ρ is X-ray mass attenuation coefficient, ρ is mass density and t is the depth from surface. Then the absorption (A) can be obtained by:

$$A = I_0 - I_t = I_0 \cdot \left[1 - \exp\left(-\frac{\mu}{\rho} \cdot \rho \cdot \bar{t}\right)\right] \qquad \text{Eqn. 3}$$

Considering that the sample is being rotated during the imaging, the thicknesses here should be the average values of the integrated thicknesses with incident angle from 0 to 360°. The value of μ/ρ and ρ for all elements can be found in [6].

According to Eqn. 2 and Eqn. 3, the X-ray absorption of common materials used in PCBs and potential materials for X-ray blocking are calculated and plotted in Figure 5. As shown, for 1.6 mm thickness (0.062 inches, the typical thickness of a six-layer PCB [Sunstone Circuits]), common PCB materials such as C, O, Al, Si, Cu, and Ge allow most X-rays to pass through, while high-Z materials including Hf, Ta, W, and Pb can block almost all incident X-rays. Even shrinking the thickness to 0.36 mm (0.0014 inches, the typical thickness of a laminate core in a six-layer PCB [Sunstone Circuits]), the high-Z materials can still block about 90% of incident X-ray (e.g., 92.5% by Ta and 95.4% by W).

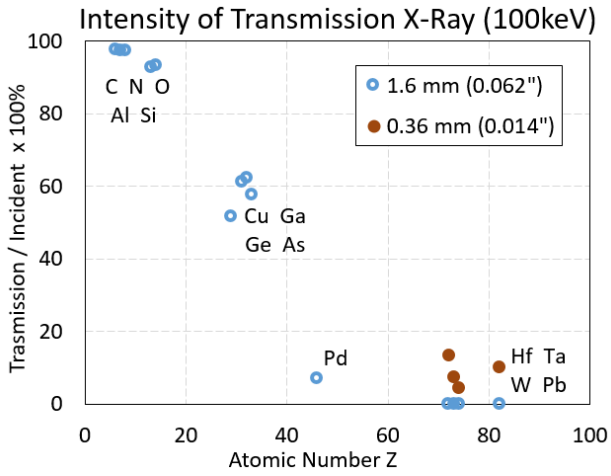


Figure 5. X-ray absorption in different materials.

Trace Permutation

The connections covered under the high-Z material are concealed from the images reconstructed using X-ray tomography. Placing this material through the whole layer is costly and impractical; however, partially covering the original traces enables the attackers to efficiently remove the mystery from the images. This mystery indicates the noise induced by the addition of high-Z material. The reason is that traditional PCB designs attempt to place the traces one after one in parallel on the same layer (shown in Figure 6 (a)). Since the traces are in the same layer, the choice of connections is unique. For instance, if the middle of the traces in the original design (Figure 6 (a)) is covered with high-Z material, one can still easily figure out the correct connections by matching the order of the traces. Thus, the traces should be permuted and routed in different layers prior to being covered with high-Z material. This process is presented in Figure 6 (a-c). In this section, the algorithms to achieve this permutation and the corresponding evaluation metrics are provided.

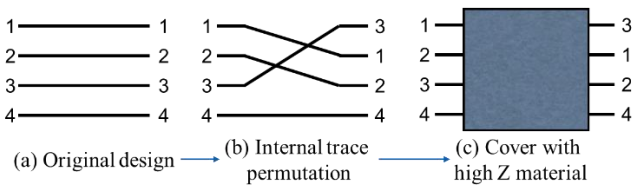


Figure 6. Permutation of the PCB metal traces.

Three parameters should be considered in implementing this permutation:

Algorithm 1

1. set $l \rightarrow 1$;
2. Total number of connections: tn
3. Randomly pick one trace (t_1) from all the connections;
4. set $L_1 \rightarrow \{t_1\}$; \\ layer #1 consists of trace t_1
5. For ($i = 2,\ i < tn,\ i++$)
6. For ($j = 1,\ j < l,\ j++$)
7. If (t_i intersects with any trace $\in L_j$)
8. $l++$;
9. $L_j = \{t_i\}$; \\ add a layer for t_i
10. else
11. $L_j = \{L_{j-1}, t_i\}$; \\ append t_i to the non-conflict layer break;
12. endIf
13. endFor
14. endFor

1) The number of input/outputs permuted (*m/n*);
2) The number of middle layers needed (*l*);
3) The trace assignment information (*L*).

The parameter *m/n* refers to the number of traces that are involved in the permutation. The parameter *l* indicates the number of middle layers that can be utilized to route these permuted traces. The assignment parameter *L* consists of the information regarding which traces should be routed in which layer. Besides these three implementation parameters, two metrics should be considered for the evaluation purpose.

1) Total number of combinations (*c*)
2) Area overhead (*a*)

The combinations refer to the total number of trials an attacker needs to find the correct connections. The area overhead indicates the amount of area induced by applying the permutation. To implement and evaluate the trace permutation, the following two goals should be accomplished:

- Goal 1: $f(m,n) = (l, L)$
- Goal 2: $f(m,n,l) = c$

Goal 1 calculates the maximal number of middle layers required to realize a fixed input/output order and the trace assignment for each layer. Goal 2 computes the maximal number of combinations based on the number of middle layers available and inputs/outputs that are involved in the permutation.

Two algorithms are proposed to achieve these goals. **Algorithm 1** generates the number of layers needed and the trace assignment information through the knowledge of the input/output order. In this algorithm, we assume the number of inputs and outputs are equal (i.e., *m* = *n* = *tn*). The case in which these numbers are not equal will be discussed later in this section. During the initialization, the number of layers *l* is set as 1 (step 1). Next, the algorithm arbitrarily picks one trace and assigns it to the first middle layer (steps 3 and 4). Then, the algorithm examines all the rest of the traces to determine whether they intersect with any traces in each middle layer (steps 5 to 15). For each trace, if this intersection is present in all the existing middle layers, another middle layer should be added (steps 7 to 9). Otherwise, this trace can be appended to the layer that presents no intersections. At the end of Algorithm 1, the number of required middle layers is reported in variable *l* and the assignment information is contained in variable *L*.

Algorithm 2 calculates the number of combinations (*c*) given the number of inputs/outputs (*m/n*) and middle layers (*l*). Similar to Algorithm 1, the numbers of inputs and outputs are assumed as equal (i.e., *m* = *n* = *tn*). The number of inputs/outputs is split into *l* subsets (step 1). For instance, if six traces are involved in the permutation (i.e., {*m*,*n*}={6,6}) and three middle layers can be used to route these traces, the possible splits are

- $s_1 = \{\{1,1\},\{1,1\},\{4,4\}\}$;
- $s_2 = \{\{1,1\},\{2,2\},\{3,3\}\}$;
- $s_3 = \{\{2,2\},\{2,2\},\{2,2\}\}$.

Algorithm 2

1. Split $\{m,n\}$ into l subsets: $s = \{\{m_1,n_1\},\dots,\{m_l,n_l\}\}$, where $\sum_{k=1}^{l} m_k = m$ and $m_k = n_k > 0\ \forall\ k \le l$.
2. Set total number of splits $\rightarrow d$
3. For ($i = 1, i \le d, i++$)
4. For ($j = 1, j \le l, j++$)) $\{m_j,n_j\} \rightarrow a_{i,j}$ \\ compute the number of assignments
endFor
5. endFor
6. endFor
7. Total combinations: $c = \sum_{i=1}^{d} \prod_{j=1}^{l} a_{i,j}$

The number in each subset indicates how many traces should be routed in that layer. The number of subsects is stored in *d* (step 2). During steps 3 to 7, for each subset, the numbers of possible assignments are computed. This process is illustrated in Figure 7. The first subset {1,1} indicates that one trace should be routed in first middle layer. Thus, 1 out of 36 connections can be selected (i.e., $a_{1,1}$ =36). Among these possible connections, three of them are present in Figure 7 (a). The second subset indicates that the second middle layer contains one trace. This trace comes from 25 possible connections (three of them is provided in Figure 7 (b)) since one trace has been routed in the first middle layer. Thus, $a_{1,2}$ equals to 25. The third subset shows that the rest of the four traces that are routed in the third middle layer. As shown in Figure 7 (c), only one possible connection (i.e., $a_{1,3}$=1) can be achieved when the traces in other middle layers are fixed. The reason is that all the traces in the same layer cannot be overlapping. The above process is repeated until all the ai,j are computed. Finally, the number of total combinations can be obtained in step 8.

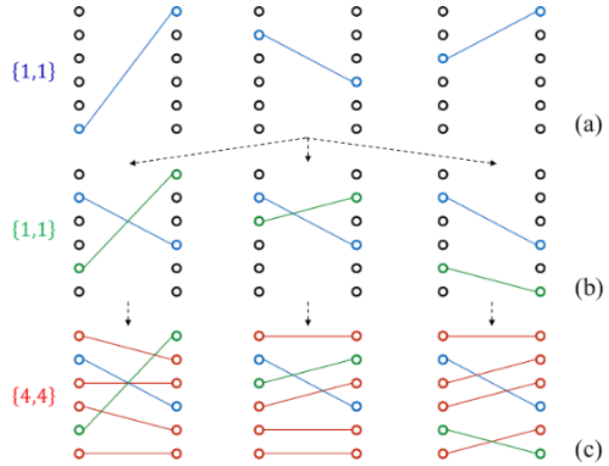


Figure 7. Possible assignments for the split $s_1 = \{\{1,1\},\{1,1\},\{4,4\}\}$.

In summary, Algorithm 2 tells a designer how difficult it is for an attacker to break through the protection under certain design constraints. Algorithm 1 provides a designer the way to route the chosen traces. However, not all the traces can be utilized in the permutation [8]. A designer should only involve the traces without dedicated functionalities (e.g., general-purpose input/outputs).

As discussed above, both Algorithm 1 and 2 assume *m* and *n* are equal. However, one can make *m* greater than *n* by involving **dummy traces** [8]. The dummy traces are the ones that are not present in the original design. If the dummy traces are engaged and $m > n$, the total combinations can be computed by the Equation 4.

$$c = \binom{m}{n} * c_n \qquad \text{Eqn. 4}$$

where c_n represents the number of combinations computed by Algorithm 2 with $\{m, n\} = \{n, n\}$. Another assumption in Algorithm 1 is that if one trace overlaps with two traces from different layers at the same time, a third layer must be added to route this trace (Figure 8 [a]). Alternatively, this trace can be routed in existing layers by adding vias (Figure 8 [b]). This action decreases/optimizes the number of layers. However, this optimization is not guaranteed since it is constrained by design rule checking (DRC).

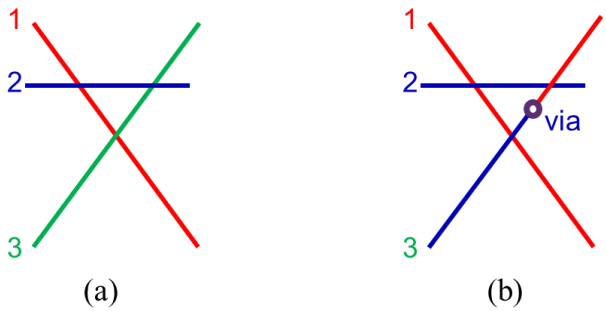


Figure 8. Number of layers optimization.

Combining Algorithm 1, Algorithm 2, and the optimization process, the following framework can be exploited to implement the permutation.

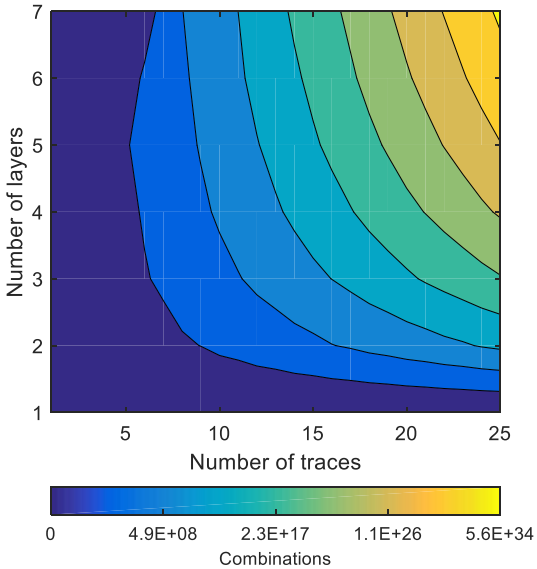


Figure 9. Parameters selection.

Step 1: The designer selects the parameters (m, n, l) based on a combinations target (c). This can be done by changing the values {m,n} and in **Algorithm 2**. According to each design, the optimal m and l can be selected. The relationship between {m,n} and l is shown in Figure 9. In this figure, the numbers of middle layers are given in the y-axis and the number of traces to permute are shown in x-axis. The color indicates the number of combinations.

Step 2: The numbers of inputs/output and middle layers are given by Step 1, and their orders can be randomly selected. These traces are transferred to **Algorithm 1** as inputs. After executing Algorithm 1, the number of middle layers needed (l') and the layer assignment information ($L_j \forall j < l'$) is given. Note that l' may be smaller than l because of the random selection of the input/output orders. If $l' = l$, the optimization can be applied. During this optimization, L_j is sorted by the number traces in each middle layer. For the middle layer that contains the least number of traces, the designer may route the traces in this layer within other layers by adding vias (Figure 8 [b]). By applying this optimization, the required number of middle layers may be reduced.

Table 1. Performance evaluation.

Design No.	Input/ output (m/n)	Middle layers (l)		Comb. (c)
		Require	Actual	Total
1	16/16	3	4	1.2e+13
2	14/14	3	0	1.7e+11
3	18/18	3	0	9.3e+14
4	19/19	3	4	7.1e+15

We apply Step 1 of the proposed framework on four reference designs (listed below) from Altium to evaluate the permutation performance.

1. DB30 Xilinx Spartan-3 FGG676
2. SL1 Xilinx Spartan-IIE PQ208
3. NBP3 Altera MAX7000 MAX3000 PLCC
4. NanoBoard-NB2DSK

This performance indicates the difficulty for an attacker to guess the true traces. This evaluation can be accomplished by computing the number of combinations (c). The target number of combinations is set to be greater than 1e+10. The results are presented in Table 1.

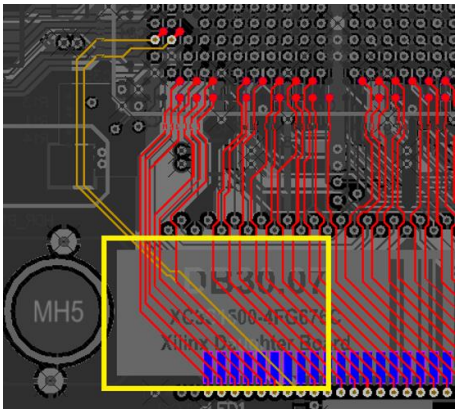


Figure 10. Permutation region of design No. 1

In this table, the Input/output column presents the numbers of traces involved in the permutation. Under the Middle Layers column, the Require subcolumn indicates the number of middle layers needed to achieve the combinations listed in the last column. The Actual subcolumn shows the number of middle layers in the original design. According to Table 1, some designs can provide a sufficient number of middle layers, which some cannot. For the ones that consist of enough middle layers, the designer can select the required number of traces and reroute them in a small region. For instance, the region where the permutation can be implemented on design No. 1 is marked in the yellow rectangle in Figure 10. However, for the other designs (e.g., design No.2 and No. 3), redesign is required to add more layers.

Area Overhead for Permutation Network

We also assessed the area overhead from implementing the proposed permutation network. Here, our goal is to implement the permutation network in the absolute least-area possible. An example of such a minimal-area routing solution (in an 8 input/8 output permutation network) is shown in Figure 11. Here, the red traces (which serve as I/O to the network) are present in layer-2. Using a staggered configuration of vias (spaced within design rule constraints), these layer-2 traces can now be routed to any of their respective layers. We can then route these traces in their respective layers while keeping the minimum trace-to-trace distance ($t \leftrightarrow t$) that the design rules allow us. The required area for this

permutation network on the PCB board would then be given by the following:

$$\text{Total Area} = \text{Width} \cdot \text{Height} \qquad \text{Eqn. 5}$$

$$\text{Width} = d_{via} + 4v_r + 2(t \leftrightarrow v) + t_w + N \cdot \left((t \leftrightarrow t) + t_w\right)$$

$$\text{Height} = (m-1) * (v_r + (t \leftrightarrow v) + 0.5t_w) + 2v_r$$

$$d_{via} = \sqrt{2v_r + (v \leftrightarrow v)^2 - (v_r + 0.5t_w + (v \leftrightarrow t)^2}$$

All the parameters in equation 5 are annotated on Figure 11. Most of the parameters such as d_{via} (via diameter), $t \leftrightarrow t$ (minimum track-to-track gap) etc. are determined by the PCB manufacturing process (through design rules).

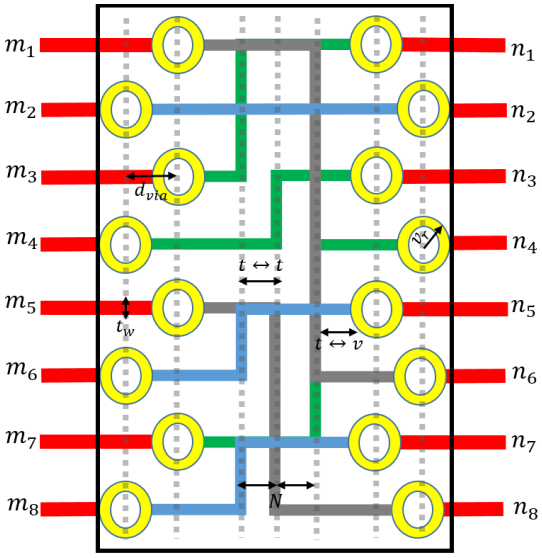


Figure 11. Manual Routing in a Permutation Network with 3 Layers

The parameter N denotes the number of track-to-track gaps required in order to realize the permutation (where N = 2 in Figure 1). A gap is created each time either of the scenarios depicted in Figure 12 occurs, where a source (or sink) node Source_i or Sink_i is lower in position than a reference node ($\text{Ref}_{\text{sink}_{i-1}}$ or $\text{Ref}_{\text{source}_{i-1}}$) that is right above the current source/sink node.

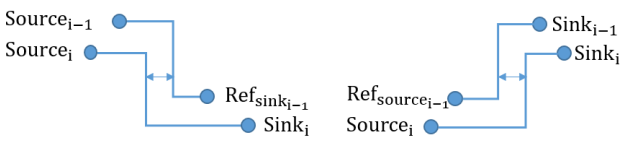


Figure 12. Track-to-Track Gap Creation During Routing

Design No.	Input/ output (m/n)	N	Area (W x H) / mm^2	% Area of Total PCB Footprint
1	16/16	5	106.41 (6.26 x 17.01)	1.14
2	14/14	4	85.60 (5.74 x 14.92)	0.94
3	18/18	6	129.40 (6.77 x 19.10)	2.31
4	19/19	5	126.05 (6.26 x 20.15)	2.55

Table 2. Area Overhead Estimation

Using this notion of minimal-area routing, we estimated the required area overhead for each of the four reference designs from Table 2 (all of which required three layers to implement the proposed permutation network). For each design, we generated several random permutations of (m, n) and determined an average N. Note that each layer has its own average value of N and the value of N for a design is the maximum value of N out of all the layers. We then used Equation 5 along with the minimum design rule values obtained from Advanced Circuits, AZ (via PCB Artist) to calculate the area. We can see that for all of the designs, the permutation network can be achieved in < 130 mm^2 of design area. This translates to < 3% of the total board area for all the designs. The low area overhead also makes it possible for the designer to strongly obfuscate the board by incorporating more than 1 of such permutation networks, if additional area is available.

Robustness Against Other Attacks

Since the attacker cannot learn the design information through X-ray based attacks, he or she may attempt to discover the correct connections by other attacks. An attacker might inject a pulse into one of the permuted connections. By monitoring the response of this pulse among all the permuted connections, the attack can reveal the correct connection. This attack can be viewed as a testing-based attack. The PCB may also experience a hybrid attack which combines this testing-based attack and the X-ray based attack.

For the PCBs which are protected by the proposed framework, this hybrid attack can barely be executed for the following reasons. In order to implement this attack, the attacker is required to precisely inject the signal into each of the permuted traces. In order to accomplish this task, the attacker can either inject the

signal (i) through the vias or (ii) through the middle of the traces which are not covered by the high-Z material. The former approach is not applicable since the vias are also blocked by high-Z material. Thus, via-to-trace relationship is concealed to the attacker. The latter approach cannot be implemented since the permuted traces are routed in the middle layers. The attacker cannot obtain access to these traces unless he or she resorts to a destructive attack (which is beyond the scope of this paper). As a result, the proposed protection framework is fully resilient to the aforementioned hybrid attack. Further, if the components mounted on the top/bottom of the board use BGA (Ball Grid Array) packaging along with routing of the connections in the middle layers, it will also not be possible for the attacker to probe the pins/connections of the components mounted on the PCB and thereby find connectivity information.

Conclusion

In this paper, a framework for protecting PCBs against non-destructive reverse engineering using X-ray microscopy was presented and analyzed. Central to this approach is the addition of high-Z material that increases noise during tomography in permutated regions of a PCB. Based on X-ray tomography noise and X-ray absorption introduced by high-Z materials, tantalum and tungsten are identified as promising candidates for this application. We have provided efficient algorithms to estimate the protection offered by the scheme (i.e., how many I/O combinations need to be tested by an attacker to identify the correct permutation), to identify various parameters (number of layers, traces, etc.), and compute how much additional area and/or layers must be consumed. Results for four commercial PCB benchmarks show high-level of protection and be obtained using the proposed approach at modest cost. We show that a large number of combinations can be obtained with modest number of internal layers (3) in a PCB. In addition, the permutations can be realized with affordable area overhead (less than 130 mm^2).

Acknowledgments

The authors would like to acknowledge support from National Science Foundation (grants CCF-1423282, CCF-1559772, and CNS-1558516) for this work.

References

[1] R. Torrance and D. James, “The state-of-the-art in ic reverse engineering,” in *Cryptographic Hardware and Embedded Systems-CHES 2009*, Springer, 2009, pp. 363–381.

[2] N. Asadizanjani, et. al., “Non-destructive PCB Reverse Engineering Using X-ray Micro Computed Tomography,” in *41st International Symposium for Testing and Failure Analysis (November 1–5, 2015)*, Asm, 2015.

[3] S.E. Quadir, J. Chen, D. Forte, N. Asadizanjani, S. Shahbazmohamadi, L. Wang, J. Chandy, M. Tehranipoor, “A survey on chip to system reverse engineering,” *ACM journal on emerging technologies in computing systems (JETC)*, 2015.

[4] J. Grand, “Printed circuit board deconstruction techniques,” in *8th USENIX Workshop on Offensive Technologies (WOOT 14)*, 2014.

[5] J. T. Cremer, Jr., *Neutron and X-ray Optics (1st edition)*, Elsvier, 2013, ISBN 978-0-12-407164-3 (Chapter 2).

[6] J. H. Hubbell and S. M. Seltzer, “Tables of X-Ray Mass Attenuation Coefficients and Mass Energy-Absorption Coefficients from 1 keV to 20 MeV for Elements $Z = 1$ to 92 and 48 Additional Substances of Dosimetric Interest,” NISTIR 5632, NIST, 2004.

[7] Department of Health and Human Services, U.S., “Public health statement – tungsten, CAS# 7440-33-7,” 2005.

[8] Z. Guo, M. Tehranipoor, J. Di, and D. Forte, ”Investigation of obfuscation-based anti-reverse engineering for printed circuit boards,” in *Proceedings of the 52nd Annual Design Automation Conference*, p. 114. ACM, 201.

[9] J. Theodore Cremer, Jr., *Neutron and X-ray Optics (1st edition)*, Elsvier, 2013, ISBN 978-0-12-407164-3 (Eqn 2.19).

[10] B. Cabral, N. Cam, and J. Foran. "Accelerated volume rendering and tomographic reconstruction using texture mapping hardware." *In Proceedings of the 1994 symposium on Volume visualization*, pp. 91-98. ACM, 1994.

[11] N. Asadizanjani, "3D Imaging and Investigation of Failure and Deformation in Thermal Barrier Coatings Using Computed X-ray Tomography." (doctoral dissertation), 2014

[12] N. Asadizanjani, S. Shahbazmohamadi, and E. H. Jordan, “Investigation of surface geometry change in thermal barrier coatings using computed x-ray tomography,” In Developments in Strategic Materials and Computational Design V: A Collection of Papers Presented at the 38th International Conference on Advanced Ceramics and Composites January 27-31, Daytona Beach, Florida, pp. 175-187. John Wiley & Sons, Inc, 2014.

[13] N. Asadizanjani, and E. H. Jordan. "Optimization and Development of X-ray Microscopy Technique for Investigation of Thermal Barrier Coating." Processing and Properties of Advanced Ceramics and Composites VII: Ceramic Transactions, Volume 252, pp. 425-440, 2015.