Achieve tamper-proof capacitive sensing

Vikram, Cypress Semiconductor - January 12, 2015

Applications such as Point Of Sale (POS) devices and keypads for secure door locks are required to be tamper resistant. If these devices are tampered with, then there are possibilities for theft of confidential information such as the Personal Identification Number (PIN) of a credit/debit card or access code of a lock. Hence, the devices incorporate special measures to detect tampering and halt further operation to avoid loss of sensitive data.

One of the easiest ways to gain access into a device is through the region where electrical contacts are brought out, or are closer to surface. Mechanical buttons are required to gather user input and these generally use tactile switches placed beneath the rubber or plastic key mat. Since switches have electrical contacts that connect them to microcontrollers and, as these switches are relatively close to surface, they are an easy target for tamper attempts.

A micro drill is used to make small holes on the keypad overlay and reach electrical contacts beneath them. Once electrical contacts are reached, very thin wires are attached to them and are connected to a parallel processing system as shown in Figure 1. The apparatus is then hidden from the view of a user. When a user operates the keypad to enter his secure code, the device scans the buttons to gather input. Simultaneously, the parallel processing system passively monitors button presses and records them. User will have no clue that the device is tampered with and thus become the victim of a tampering attack.
Similarly, there are other numerous ways used to tamper with a design. These may include tapping the communication lines of the controller or monitoring the transactions of the device passively by fixing a ‘bug’ on the system. It is also possible to cut the power supply of the device and tamper the system. The component of the system, which has least measures to protect it against tampering, becomes the most vulnerable part and thus the target of tampering attacks. Hence, it is not sufficient if we incorporate measures to secure any single component of the device in order to make the whole system tamper resistant. The combination of anti-tamper measures incorporated on every part that goes inside the device is what makes the system tamper-proof. In this article, we will concentrate on how to make the keypad tamper resistant.

To detect when there is an attempt to drill through the keypad, generally a very fine wire mesh is placed on switches just beneath the plastic keypad overlay. The mesh is a single contiguous wire with two ends; the wire loops and circles around to form a protective layer on the switches. A fixed voltage is sent at one end of the wire mesh and it is measured at the other end by an Analog to Digital Converter (ADC). An attempt to drill through the button overlay to tap into the electrical contacts results in breakage of the wire mesh as it is made of a very fine wire. Hence, the voltage read at the other end of the mesh will no longer match the expected value. This indicates that the mesh has been broken and possibly the device has been tampered with. Further transactions are halted and the device stops all operations, hence avoiding theft of confidential data.

We will also have to take care that it is not possible to open up the button keypad and remove the wire mesh, otherwise measures must be incorporated to detect that it has been opened and hence tampered with. For example, to detect the removal of the keypad button overlay, additional electrical contacts are made on the rubber key mat and at corresponding places on the PCB underneath it as shown in Figure 2. A current is sent periodically through these electrical contacts to check continuity. When the rubber keypad is removed from the PCB, the current flow on the electrical contacts breaks and the tampering attempt can be detected.
In yet another method, to prevent the insertion of external components inside the device, a capacitive sensor is placed as shown in Figure 3 and the capacitance on the PCB is monitored regularly. Capacitance measured by the sensor changes when a foreign material is introduced into the device while trying to open it, or if a bug is placed in the device. The external components add additional capacitance, which can be detected by the capacitance sensor and the tamper attempt can thus be detected.

Likewise, numerous measures are used to make the keypad tamper resistant. But for a design to be commercially viable, the number of components used and manufacturing cost has to be kept low, as every component used must be ensured that it cannot be tampered with. Most anti-tamper techniques require additional components or sensors, which increase the cost of the design. So if we keep the component count low, we reduce the cost of the design.

Let us see if capacitive touch buttons suit these requirements. A capacitive touch button consists of a copper pad in the form of a circle or square connected to a microcontroller with a trace. An overlay such as a layer of glass or acrylic is glued firmly on top of the buttons, which makes it very difficult
to open up and remove the overlay as such. A mechanical button assembly on the other hand consists of a rubber or plastic keypad overlay, tactile switches beneath the keypad overlay, and additional measures to prevent removal of keypad overlay. Capacitive touch buttons require fewer components than mechanical buttons and are hence cost-effective. Next we need to incorporate measures to detect if there is an attempt to tamper with the touch buttons.

The wire mesh, which provides security against tampering for mechanical buttons is placed on top of tactile switches, just beneath the overlay. But if we place the same wire mesh over the capacitive touch buttons, they can no longer detect a finger touch. This is because the metal wire mesh forms a cage for the electric field lines. Hence the field lines emitted by the touch buttons cannot propagate through this metal mesh. But the wire mesh is required to provide security against tampering. In order to accommodate the mesh in a touch button design, we can redesign the buttons as discussed below. **Touch Button Pads**

A touch button generally is in the form of a solid metal pad as shown in Figure 4(a). The metal pad can be in the shape of a circle or a rectangle or a square or any other solid shapes. Instead of using a solid metal fill, we draw the metal pour into a very fine line and loop it as shown in Figure 4(b). The loop is drawn big enough such that the touch button gets sufficient contact surface for a finger touch.

![Figure 4](image.png)

*Figure 4. Conventional solid touch button pad on the left and non-solid pour button formed by looping of metal trace on the right*

We surround the touch button trace with another fine metal trace, referred to as a shield trace hereafter. The shield trace runs in parallel to and very close to the touch button trace and on either side of the touch button trace as shown in Figure 5(a). The distance between the touch button trace and shield trace on either side is kept very small so that an attempt to drill a hole will damage the shield trace. In other words, it is not possible to gain access to the touch button trace without damaging the shield trace since they are intertwined together very closely.
Similarly, we make other touch buttons of the keypad by drawing the touch button trace and the shield trace next to each other. The shield trace used for different touch buttons is a common trace. That is, the shield trace after it encircles a touch button continues further and encircles other adjacent touch buttons as shown in Figure 5(b). The spiral pattern of winding the touch button trace and shield trace shown here is for explanation purpose only. The pattern of drawing touch button trace and shield trace in actuality is made as a maze so that it is very difficult to distinguish them visually apart as shown in figure 6. The mesh formed by shield trace is referred to as a security mesh hereafter.

The touch buttons present on the top layer of the PCB need to be routed to a microcontroller, which is generally present on the bottom layer. Vias are used to connect the top and bottom layers of the
PCB and route the touch button traces. But these vias are also susceptible to tampering as one can tap them and access the signals being routed to the microcontroller. Hence, the vias of touch button traces are buried in inner layers of PCB to make them inaccessible from the outside. The touch buttons embedded in the security mesh layer with overlay on top and the controller present in bottom layer of POS device is shown in Figure 7.

![Figure 7. POS device with overlay layer glued to the capacitive touch buttons intermixed with the security mesh](image)

Placing a metal trace close to the touch button’s trace, the way we have done in this design, increases the self-capacitance of a touch button. Increased self-capacitance results in lower sensitivity to finger touches. This is because the controller will have to detect a relatively small increase in capacitance due to a finger touch on a button of relatively larger capacitance. To reduce such parasitic effects of capacitance, PSoC provides the option of a signal shield. The shield signal is a buffered version of the voltage on a touch button that can be driven on a security mesh. When both the touch button trace and the security mesh are at the same voltage, the capacitive coupling between them is reduced. But the same wire mesh also needs to be used for security scanning, where we confirm the contiguity of the mesh. To switch between the voltages of shield signal during button scan and security scan signal during a tampering check, we use an analog mux present in the PSoC. The shield voltage buffer is connected to the analog mux and this drives the pin connecting the security mesh as shown in Figure 8.
**Enabling Security**

The other end of the security mesh is connected to one more pins of the PSoC and a fixed resistance connected to ground is connected to the same pin. The shield trace and fixed resistor form a voltage divider at the pin where both connect to PSoC. While one end of security mesh is driven with the shield signal, the other pin is tristated (high impedance state) by changing the drive mode of the pin. Touch buttons are scanned one after other by connecting each button to a Capacitance to Digital converter (called a CapSense block in PSoC). The pattern of scanning touch buttons is made to follow a pseudo random order instead of the conventional round robin sequence. This increases the effort required to decode sensor signals if someone tries to tap the signal on the touch buttons. Designs in which power consumption is a concern can connect the grounded terminal of resistance to a pin instead and tristate this pin when scanning touch buttons. This avoids the static leakage of current through a resistor to ground when the shield signal is driven on it. The pin can be driven to ground when scanning the security mesh for contiguity.

After finishing scanning of all touch buttons, the buffer driving the security mesh is disconnected from the analog mux. We now connect a DAC (Digital to Analog Converter) to the same analog mux and a known voltage is driven across the security mesh. The Capacitance to Digital converter is now disconnected and an ADC is connected to the analog mux. The pin, which was tristated earlier, is connected to this mux so that the ADC can measure the voltage on this pin as shown in Figure 9. The analog mux of a PSoC makes it very convenient to make and break connections on the fly. Any general purpose pin can be connected to an analog mux on one side and on the other side we can connect an internal component which can either drive the pin, such as a DAC or a component which can sample the voltage on pin, such as an ADC or Capacitance to Digital converter. Specifically in this case, we are using the mux to switch between driving of shield voltage and DAC voltage on the security mesh.
The voltage $V_{\text{input}}$, sent at one end of the mesh drops along the resistance of maze network of security mesh. Voltage $V_{\text{output}}$ on other end of mesh, the junction where it connects to ADC and resistor can be calculated for a given $V_{\text{input}}$ as we know the effective resistance of the mesh network beforehand. This value is compared against the voltage read by ADC. An attempt to drill the keypad to gain access to the button traces will result in breakage in contiguity of the security mesh. The voltage at the other end of the mesh read by the ADC no longer maintains its expected value but becomes floating due to the discontinuity. This indicates an attempt to tamper with the keypad and all further operations are halted in the device. We can vary the voltage driven by the DAC during each security mesh scan, following a predetermined sequence of fixed voltages. This makes it difficult for someone trying to access the security mesh and mimic driving a voltage on it, and hence considerably increases the effort to tamper.

The controller alternately scans the keypad touch buttons and the security mesh. Once the security mesh contiguity check fails, it indicates a possible tampering attempt. Further transactions on the device are halted to prevent the theft of any confidential information. The security mesh can be scanned after scanning of all the touch buttons. If it is required to increase the security, then the mesh can be scanned after every touch button scan. Likewise, more features can be incorporated to increase security against tampering.

This approach is able to reasonably detect when a system is being tampered with and halt the system, thereby avoiding theft of secure data. As mentioned earlier, the measures mentioned in this article provide tamper-sensing capabilities to a keypad. However, we need to remember that this by itself is not sufficient to make the entire device tamper resistant, but a host of anti-tamper features for every component which goes into the design needs to be incorporated for a successful tamper proof design.
Additional Information: [Getting started with CapSense Guide](#)