THE INDIA FORUM
A JOURNAL-MAGAZINE ON CONTEMPORARY ISSUES

January 1, 2024

Bringing Trust to Electronic Elections in India

By: Madhav A. Deshpande

EVMs cannot be hacked but they can be compromised. The key to digital election trust is a complete audit trail. Every EVM needs a tamper-proof log that shows every power blip, button press, and message, all time-stamped. This we do not have now and is a concern for voters.

Indian elections are electronic now. We must accept that Indian electronic voting machines (EVMs) cannot be hacked. It is a fact. However, we must recognise that unlike stealing a ballot box, which is a very noticeable, physical act, EVMs can be compromised (not hacked), completely unnoticed and on a national scale.

Many people and organisations like the Citizens' Commission on Elections have raised many issues about EVMs. Important concerns pertaining to the Voter Verifiable Paper Audit Trail (VVPAT) have been raised by Gopinath Kannan in an article published in *The India Forum* on 13 April 2021. He has elaborated on the risks entailed in allowing personal computers (PCs) to connect to the VVPATs days before the polling date to upload candidate names and symbols and link those to the voting buttons using a "symbol upload module". As a computer scientist, I believe that such interjection has the potential to go much beyond mere key-mapping.

Here, I discuss the following issues.

1. It is impossible to guarantee that the program code is untainted.

2. The only way to guarantee that the VVPAT does not tamper with votes is by tallying the copies of electronic votes before the VVPAT and after the VVPAT—that is, copies of votes in the ballot unit (BU) and the control unit (CU) must be identical.

3. Incremental, progressive checksums of votes in the BU and CU must be taken after every vote is received and processed in the BU to the CU. (A checksum is a combination of letters and numbers that results uniquely for a given set of data. Even the smallest change in data results in a different checksum. Checksums are globally used to confirm if anything has changed.)

4. Programs can start to behave differently on triggers like specific key presses or date and time. A full, unalterable, retraceable audit trail is the only way to provide undisputable proof of fair play.

5. A fully retraceable and unalterable log (audit trail) must be maintained by every EVM. It must include every event from the EVM's first power-on till the counting of votes to prove the integrity of every vote beyond doubt.

6. Every EVM must include a battery-backed, half-duplex Global Positioning System (GPS) transmitter sharing its location with a satellite at every moment.

> Given that many people, organisations, and political parties are involved, it is impossible to expect that the electronic election process will become and remain a fully trusted closed system at any time.

An electronic election (EE) process has many leakage points, just as any other public process. Every partially closed system needs post-facto, confirmatory, non-immutable evidence of sequential events.

We must therefore establish mechanisms of verification and audit that will address any and every possible claim of discrepancy and thus build total transparency and trust. Such a mechanism will be continuously "on", but will be called upon post-facto—that is, after the voting has taken place.

Establishing a robust and transparent system for an electronic election is the duty of the government, the Election Commission, and the Supreme Court (SC). Failing to do so is a direct denial of the right to franchise to every Indian citizen. Protecting the right to franchise means making sure that every vote remains unaltered and secret. All citizens have a right to know how the electronic election system protects their right to franchise from being misused. Such information must not be kept secret under the garb of "national security". There is absolutely no reason for any honest citizen to believe that someone else is more trustworthy and agree with that

person just because he or she occupies a particular seat. Everyone has an equal right to evidence.

It is impossible to guarantee that the program code is untainted. Even if the program code was in the public domain, what is the guarantee that the code that has gone into the device is the same as the one in public view? Just a change in few lines of code can completely alter the behaviour of a device. And those few lines can be changed just a few moments before the code goes into the device.

Even the most honest person's integrity can be forcibly compromised by simple utterances such as "we know your kids go to xxx school" or "we know your parents/wife/brother". Right?

> The VVPAT has introduced another non-trivial challenge. The vote is no more cast by the BU. Because it goes via the VVPAT, the VVPAT, not the BU, casts the vote into the CU.

The Election Commission claims that two copies of electronic votes are maintained. It is unclear when and where the copies are created. How is it ensured that both copies are identical and remain identical all the time? Which of the two copies is used in counting?

The VVPAT has introduced another non-trivial challenge. The vote is no more cast by the BU. Because it goes via the VVPAT, the VVPAT, not the BU, casts the vote into the CU.

Two issues concern the VVPAT:

1. The VVPAT machine may alter the vote after it is displayed to the voter.

2. The VVPAT machine may show only one copy of the vote to the voter while the second copy, which may or may not be identical to the one shown to the voter, passes to the CU opaquely.

What is the guarantee that the vote is unchanged after it is displayed on the VVPAT? Simply put, how is it guaranteed that the vote entering the VVPAT is the same as the vote leaving it?

If the copies are created in the BU and *if they are not identical*, only one is shown to the voter while the other passes through the VVPAT opaquely. *Does the VVPAT remain relevant in such a case?*

A progressively hashed record of the number of votes in both copies of the BU and the number of votes in both copies of the CU will factually establish if any changes were made to any of the four copies of the electronic vote. A hashed total is an alpha-numeric set of characters when a mathematical formula is applied to a data set. To make the hash-sum even more fool proof, one may include the date-time stamp along with the value of the vote stored. It would be impossible to beat such a hash-sum and alter vote values.

A standard process of progressive checksums is used by electronic record-keeping to ensure the integrity of data at all times. This being a deeper technology aspect, I am not undertaking that discussion here. A checksum is a combination of letters and numbers that result uniquely for a given set of data. Even the smallest change in data results in a different checksum.

In addition to the progressive hash-sum, progressive checksums must be maintained at definite intervals that include all the copies of votes in the CU and stored in the write-once memory in the CU. An identical process must be adopted at the same time intervals in the BU. Checksums must be maintained internally in a memory that can be written only once. Using "write-once" memory eliminates all editing.

> Programs can start behaving differently on a trigger. Triggers can be of any kind, such as multiple concurrent key presses, date and time, and so on.

We ought to keep in mind that it is not enough to demonstrate compliance only during testing. The mechanism must provide confirmations for every vote cast in the field. In other words, the passage of every vote must be retraceable on a relevant sub-system of the EVM (BU, VVPAT, and CU) from the moment a button is pressed on the BU to the moment the vote is counted from the CU.

It is obvious that such a log will highlight discrepancies at every stage. For example, inconsistencies in values of a variable directly reflecting the electronic vote (indicating that the electronic vote was changed), inconsistencies of time, and so on.

Consider these examples:

1. A Windows PC responds to the "Ctrl+Alt+Del" combination differently because it is programmed to do so.

2. Recall that on 1 January 2000 programs that were working fine until then began misbehaving. (The Y2K problem was primarily due to the date subtraction to know time lapsed, but it is important to note that the trigger was time.)

3. The interest-posting program in banking is written to behave differently on different dates. It may deposit the interest in a related account at the end of quarter whereas it may compound it in the same deposit account on a usual month-end depending on the instructions on the deposit.

When we recognise this important technical aspect, we realise the uncertainty about an EVM's misbehaviour on a given day will always exist because of the possibility of an unknown, programmed trigger linked to time. In other words, an EVM program may be built to behave differently only on/after a certain date and time.

Here are some examples:

1. Imagine a simple program like "If it is 11 am on 1 February 2023 or later, execute instructions in para 1, otherwise, execute instructions in para 2". Any number of tests before 11 am on 1 February 2023 will never demonstrate the behaviour listed in para 1.

2. As another example, imagine a program like "If path-to-take=1, execute para1, otherwise execute para2". Consider that the value of "1" is given to "path-to-take" only if it is "11 am on 1 February 2023 or later" and if "keys 1, 3 and 5" are pressed together. Instructions in para1 will be executed on any day after 1 February 2023 if anyone presses keys 1, 3 and 5 together. This "knowledge" may be with only a few and therefore could be used to their advantage.

The conditional instruction may not be present in the published program but may exist in the code put in the EVM. Remember, those all-important few lines can be changed just a few moments before the code goes into the device.

The need of the retraceable, full audit trail is therefore obvious, immediate, real, unavoidable, mandatory, and significant. It is the only way to establish beyond doubt if a device has misbehaved or not.

Every EVM must create and maintain a full audit trail. A record of every event in the EVM's life, from power-on and power-off to key presses, memory storages, messages exchanged, and so on, all with an unchangeable date-time stamp, must be maintained.

> It is important to note that in none of the hundreds of complaints received and addressed so far has the Election Commission used the audit trail to prove that nothing was wrong with the EVM …

If the Election Commission claims that audit trails are included, its response to my Right to Information (RTI) application (#24527) pointed me to a document that only mentions "event recordings" in passing, certainly giving the impression that only a few events may be recorded and a full audit trail is possibly not maintained. Leaving doubts aside, it is important to note that in none of the hundreds of complaints received and addressed so far has the Election Commission used the audit trail to prove that nothing was wrong with the EVM and the EVM was entirely and completely above board, beyond any suspicion.

Besides, every EVM must have a battery-backed GPS transmitter sharing its location every moment with a satellite. The satellite, in turn, must record this data in a cloud storage, ensuring that the geolocation of every EVM is known at every moment. The GPS transmitter will not receive any signal but will only transmit its location to the satellite. A device that only transmits cannot be hacked because it does not receive and recognise any signal or instruction from outside. It therefore cannot be ordered from outside to misbehave.

Any concerns that a GPS transmitter could be "compromised" get immediately alleviated by using a "half-duplex" GPS unit. Such a unit can only transmit the location data but cannot receive any data. It just does not have the ability to receive any data at all.

"GPS spoofing" is another concern to mention. An intermediate receiver is necessary for "spoofing" to happen. Spoofing happens when a receiver is made to believe that the source is different than the actual source. For this to happen, something must intercept the communication and re-send it after masking the original identity. An easy and accepted way to negate chances of spoofing is to send the geolocation to more than one receiver simultaneously. Unless all the data paths to the receivers are compromised in the same way, inconsistency in the geolocation is immediately caught, thereby highlighting the need of attention to the source in question.

> The most deterministic knowledge of every EVM's whereabouts can be provided by incorporating a "transmit-only" GPS module. And such deterministic real-time data must be available with the Election Commission at all times.

As many as 19 lakh EVMs have been found to be missing and the Election Commission has not been able to come clean on their whereabouts. Aside from the clear and present risk to every upcoming election and perhaps even for the elections conducted after the discovery of the loss, it is unacceptable even as a trust issue.

The most deterministic knowledge of every EVM's whereabouts can be provided by incorporating a "transmit-only" GPS module. And such deterministic real-time data must be available with the Election Commission at all times.

Implementing just these two foundational functionalities will answer the following questions that must be answered to establish complete trust in the electronic election system.

1. What is the electronic proof of fair play?

2. Do we have built-in audit trails inside every EVM to establish beyond any doubt that all events in the EVM were consistent, true, fair, and unbiased?

3. Do we tally both copies of votes in the BU with both copies in the CU?

4. How is sanitisation of thousands of PCs (that connect to the VVPATs) maintained? Are records like hash sums/checksums of the PCs maintained and when are they verified?

5. How is geolocation of every EVM recorded at every moment from the end of polling to the beginning of counting to ensure that no EVM goes off its designated track?

The reader can visit any of the links below for a demonstration and discussion.

In English: https://youtu.be/fqvt42ecr30

In Marathi: https://youtu.be/zkar4vIja2U

In Hindi: https://youtu.be/H4V7V99fK_c

Madhav A Deshpande has over 40 years of experience in the field of computer science and its applications. He holds patents like on the extension of the ISCSI protocol and he has developed the complex algorithm for US government. He has also worked as a consultant to federal, state and local governments in the US. He has architected several path-breaking software for the first time in the world, including on low-level storage management from mobile phones; a working prototype of the coveted pNFS file system etc. He has also been the CEO and CTO of many companies in India and the US.

Appendices:

A sample re-traceable trail of a vote stored internally in an EVM would look like the following in human readable language.
When a user presses the second button, say at 11:03:20 hours on 1 January 2023, the BU CPU will record: button 2 pressed at 11:03:20@01.01.2023.
This "vote" (EV) will be saved in the memory and the BU CPU will write the next record as: "2" stored in memory location 1020 on 11:03:20@01.01.2023.
If a copy of this "vote" is created, the BU CPU will next record: "2" stored in memory location 2020 on 11:03:20@01.01.2023.
The BU CPU will then construct a message to pass this electronic vote to the VVPAT.
The BU CPU will record: "2" and "2" sent to VVPAT#1234567 on 11:03:20@01.01.2023.

Upon receiving the message, the VVPAT CPU will record: “2” and “2” received from BU#2345678 on 11:03:20@01.01.2023.
Upon displaying the electronic vote, the VVPAT will record: “2” displayed on 11:03:21@01.01.2023.
Followed by: “2” and “2” sent to CU#3456789 on 11:03:22@01.01.2023.
The CU will record: “2” and “2” received from VVPAT#1234567 on 11:03:22@01.01.2023.
The CU will record: “2” stored in memory location 1020 on 11:03:23@01.01.2023.
The CU will record: “2” stored in memory location 2020 on 11:03:23@01.01.2023.

Every other event like power-on or power-off will be recorded by each subsystem in a similar manner.
(The numbers after the # are the unique identifiers of the BU, VVPAT, and CU; the CPU records of each sub-system are in their private memory.)