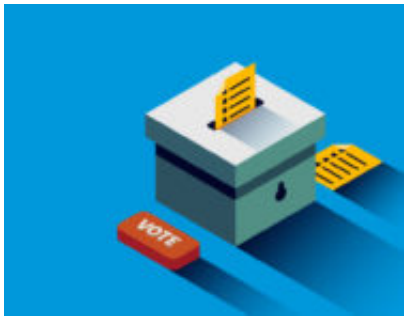
Riflessioni sulla “indipendenza dal software” nei sistemi di e-voting (parte prima)

Author : Michelangelo De Bonis

Date : 19 Marzo 2019



Introduzione

Il presente articolo vuole essere un punto di ripresa della discussione sulla tematica dell’e-voting che molto tiene banco in questi tempi. Per questo motivo proponiamo una rilettura liberamente basata sulle considerazioni basilari raccolte in un articolo scritto da Ronald L. Rivest[1], uno dei più importanti crittografi statunitensi e che riteniamo particolarmente attuale. Il lavoro più noto di Rivest è il sistema di crittografia asimmetrica, progettato insieme a L. Adleman e A. Shamir, che prende il nome **crittografia RSA** nel 1978. La sua attività di ricerca non è limitata all'ambito accademico: infatti Rivest ha avuto un ruolo significativo nel dibattito politico-sociale fra le opposte esigenze del diritto alla tutela dei dati personali (privacy) da parte del cittadino e l’esigenza del controllo della sicurezza da parte dello Stato. L’articolo da cui prende spunto il nostro intervento è un punto importante che Rivest ha messo nella discussione sull’e-voting, offrendo una visione che possa chiarire se esiste una possibilità di realizzare sistemi elettronici per il voto che garantiscano l’anonimato, la sicurezza e l’integrità del voto stesso.

Iniziamo subito dall’analisi delle considerazioni finali dell’articolo per cui la capacità di dimostrare la correttezza del software che gestisce il voto elettronico diminuisce rapidamente mentre il software diventa più complesso. È **impossibile** verificare adeguatamente che i sistemi di voto siano totalmente privi di difetti o assicurare l’assenza da eventuali bug che permettano frodi. Su questi sistemi rimane quindi sempre l’alone del sospetto di una loro incapacità di supportare elezioni sicure e accurate. Altresì sembra impossibile garantire l’effettiva indipendenza del voto dal software usato per registrarlo. Nel voto tradizionale è sempre possibile verificare la congruenza tra il voto espresso (sulla scheda) e quello registrato (durante lo spoglio) mentre nell’e-voting l’atto di acquisizione del voto e la sua registrazione si basano su meccanismi invisibili e complessi e quindi sulla fiducia che si ripone nel software stesso, dato che verificare e assicurarsi che i voti siano registrati con precisione è difficile e costoso. Il problema principale è fornire garanzia di sicurezza, integrità, riservatezza e anonimato del voto espresso: il software potrebbe essere corretto ma convincersene - o convincere tutti gli elettori -

è certamente compito arduo.

I software nei sistemi di votazione sono complessi

I sistemi di votazione elettronica sono complessi e continuano a crescere nella loro complessità. I requisiti per la privacy dell'elettore, per la sicurezza del voto contro un attacco informatico o contro il fallimento del voto stesso e, non in ultimo, l'accuratezza del conteggio finale sono tutti requisiti totalmente in conflitto tra di loro. Questi requisiti, complessi e conflittuali, conducono a una crescente difficoltà di verifica dell'affidabilità.

Un esempio riportato da Rivest riguarda i sistemi di votazione Direct-Recording Electronic (DRE), che generalmente forniscono solamente l'interfaccia utente touch-screen per effettuare la selezione del voto e che memorizzano i record di voto sia in memoria interna che su memoria rimovibile. Un DRE può visualizzare una varietà essenzialmente infinita di diversi tipologie di voto e può includere complesse caratteristiche di accessibilità per i non vedenti (ad esempio, l'uso di cuffie per essere guidato ad effettuare la scelta). Nonostante questa grande varietà di funzionalità un DRE non è in grado di offrire una verifica, direttamente osservabile dall'elettore o verificabile in fase di spoglio, che il voto sia stato correttamente memorizzato: bisogna fidarsi della tecnologia.

Il cuore della riflessione di Rivest è appunto qui: come fornire garanzie, a dispetto della complessità del software, che il sistema di voto registrerà accuratamente le intenzioni dell'elettore?

La necessità di un punto di vista diverso

È pacifico per chi si occupa di sviluppo software che la complessità sia nemica della sicurezza e dell'accuratezza: è molto difficile valutare un sistema complesso. Un errore molto piccolo (pensate a un errore di battitura nel nome di una variabile o la sua mancata inizializzazione) in un sistema complesso di grandi dimensioni può causare risultati imprevisti, in momenti imprevedibili. Oppure, può mostrare il fianco inserendo una vulnerabilità che può essere sfruttata da un avversario politico per avvantaggiarsi.

L'individuazione di tutti gli errori in un sistema di grandi dimensioni è generalmente ritenuta impossibile o estremamente costosa. *"La capacità di sviluppare software complessi supera di gran lunga la capacità di dimostrarne la correttezza o di testarla in modo soddisfacente entro limiti economici ragionevoli"*, afferma Rivest. Un sistema di voto per il quale l'integrità dei risultati elettorali dipende intrinsecamente dalla correttezza del suo software sarà sempre ritenuto per lo meno sospettabile.

L'idea introdotta da Rivest per affrontare questo problema fondamentale è quello di **cambiare ottica** e seguire una metodologia indipendente dal software: *"Verificare i risultati delle elezioni, non il sistema di voto."*

Con l'approccio DRE, si è costretti ad assumere che il software sia totalmente corretto. Se

sorgono domande a posteriori sull'accuratezza dei risultati elettorali (ad esempio è richiesto un nuovo conteggio), non c'è altra strada che quella di presumere che il sistema di voto abbia effettivamente registrato i voti in modo accurato. È chiaro, quindi, che sarebbero da preferire sistemi di voto in cui l'integrità del risultato elettorale non dipenda dalla fiducia nella correttezza del software. Non ci dovrebbe essere nessun motivo di preoccupazione che il risultato elettorale sia stato influenzato o determinato da qualche bug del software o peggio, ad esempio, da un codice malevolo esterno.

L'indipendenza dal software

Rivest definisce quindi un nuovo concetto, quello di "indipendenza dal software", che coglie la caratteristica necessaria di fornire risultati elettorali che siano verificabili, senza dover dipendere dal presupposto che il software sia corretto.

"Un sistema di votazione è indipendente dal software se una modifica o un errore non rilevato nel suo software non può causare una modifica o un errore ***non*** *rilevabile in un risultato elettorale".*

Per contrasto, quindi, si può definire un sistema di votazione dipendente dal software come vulnerabile a errori di programmazione non rilevati, codice dannoso o manipolazione del software, in modo tale che la correttezza dei risultati delle elezioni dipenda dalla correttezza stessa del software.

Per illustrare la logica dell'indipendenza dal software, Rivest propone degli "esperimenti mentali" riprendendo la tradizione iniziata da Einstein. *"Immedesimiamoci nei panni di un avversario politico"* scrive Rivest *"e immaginiamo di avere il potere di sostituire segretamente qualsiasi software esistente utilizzato dai sistemi di votazione elettronico con una loro versione appositamente modificata. Con questa precisa capacità, si può (come avversario politico) cambiare un risultato elettorale o indirizzare un'elezione senza paura di essere scoperti?"*

Se la risposta è affermativa. allora per Rivest il sistema è **dipendente** dal software; in caso contrario, il sistema è **indipendente** dal software ovvero il sistema di votazione nel suo insieme (compresi i componenti non software) ha una ridondanza e un controllo incrociato per cui un suo comportamento incoerente può essere rilevato. L'individuazione potrebbe avvenire da parte dell'elettore (che verifica il voto con un sistema incrociato), da un funzionario elettorale o da un tecnico, da un audit post-elettorale, da un osservatore esterno (in realtà va bene qualsiasi persona o ente, tranne l'avversario stesso).

Messa in questi termini il pensiero va subito a un contesto di cyber security e a interventi di hacker *blackhat*. Ma lo stesso Rivest, correttamente, astrae il contesto specificando che si può considerare un "avversario" lo stesso processo di sviluppo e test del software. Questo avversario astratto sarebbe rappresentato da tutti gli errori del software presenti fin dall'inizio e che siano sfuggiti ai processi di controllo e certificazione.

Dato che i software complessi sono difficili da scrivere e da testare, è molto probabile che conterranno numerosi "bug" non intenzionali e che a volte possano far sì che riportino risultati

errati nelle elezioni. Non è ragionevole sperare di realizzare software privi di bug, a causa principalmente dei costi proibitivi che una tale operazione sottintenderebbe.

Nella **seconda parte** di questo articolo andremo ad approfondire come in concreto la definizione offerta da Rivest può darci una soluzione a garanzia dei processi di e-voting.

Note:

[1] Ronald L Rivest, On the notion of ‘software independence’ in voting systems. Published: 06 August 2008 - https://doi.org/10.1098/rsta.2008.0149.

Articolo a cura di **Michelangelo De Bonis** e **Matteo De Simone**