# KOA: Formal Verification of Electronic/Remote/Internet Voting

## Introduction

❑Formal methods play an import role in development of critical software. In formal methods we are eliminating error prone development procedures by introducing formalized, mechanically verifiable structures

❑ A handful of votes can change the outcome of an election; every vote must be counted correctly.

❑Existing electronic voting systems are less secure than paper based systems

❑Verified Voting systems have extra security features such as the ability to prove or disprove that a vote was counted correctly

❑Need to verify the whole election procedure (not just the software/hardware) and run a verifiable election

❑Manual paper based systems are error prone, but the weaknesses are well known and understood, as compared with electronic voting systems

❑Electronic voting fraud could be disguised by technical problems

❑Electoral fraud can be very hard to detect or prove - an election is assumed to be fair unless proven otherwise

❑Remote/Internet voting have additional security risks, but usually only affect a small minority of votes and have stricter penalties for fraud

## Known Problems

Where are the weak points in the voting process?

❑ Existing cryptography is strong enough, provided that the implementation can be proven to be error free; many commercial systems are known to have bugs in their security algorithms

❑Electronic voting booths have been known to alter votes during the casting of the vote

❑Ballot layout attacks can be designed which circumvent paper-based auditing

❑Some commercial database systems have flawed database schemas; a minimum of second normal formal (2NF) is recommended to ensure consistency of the data

❑Audit trails can be exploited to find out who voted when and matched to the timestamp of each vote

❑Preferential votes can contain unique combinations of digits which could identify each voter if the entire list of votes is published

❑ Denial of service attacks (similar to the problem of filtering spam)

❑Uncontrolled software updates are permitted

❑Wireless communication through a side-channel

❑Chain of custody for voting machines

❑Chain of custody for ballot records

## Case Studies

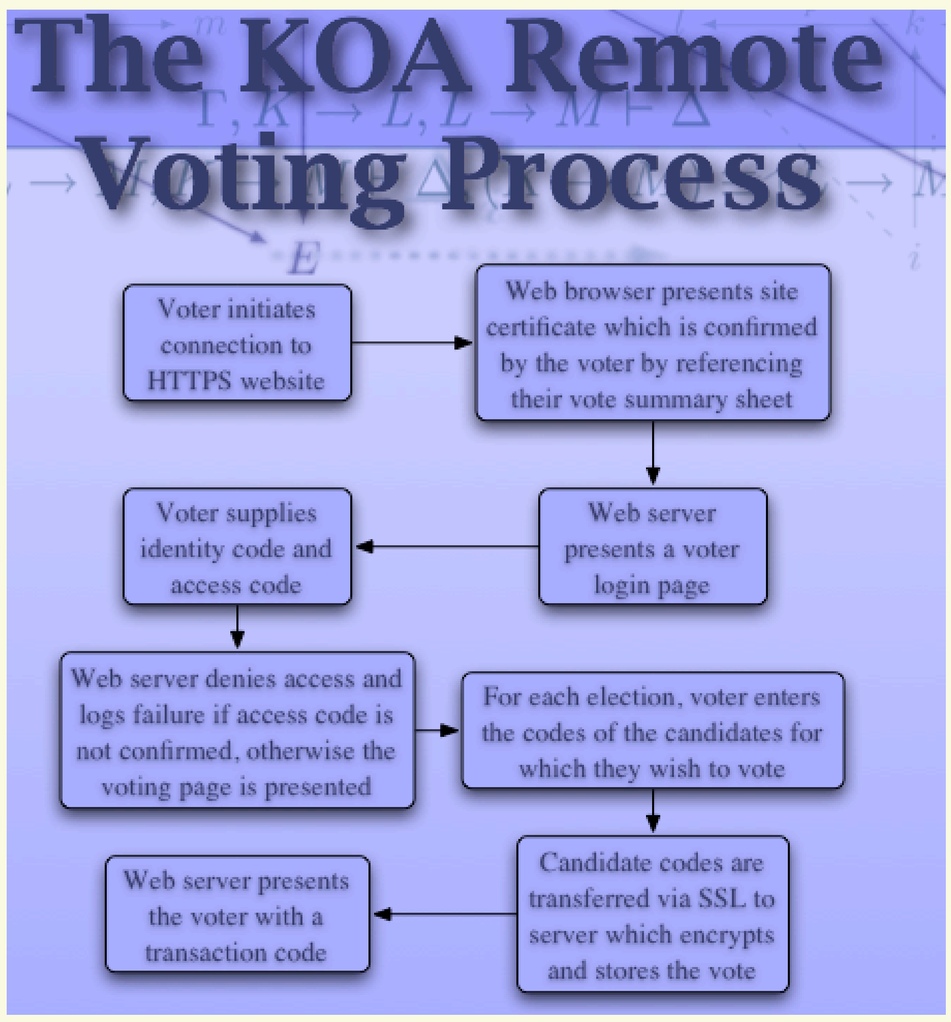
### KOA Remote Voting

❑ KOA homepage is: http://kind.ucd.ie/products/opensource/KOA/

❑Dutch and Irish vote counting algorithms have been formally specified using JML and implemented using Java (as extensions for KOA)

❑The source code for KOA is publicly available (FLOSS/open source)

❑KOA is a platform for student elections and for experiments in secure voting - it is not intended for use in governmental elections




## Work-In-Progress

Missing features of KOA which will require collaboration with other research groups…

❑VVPAT subsystem (KOA currently has VVAT without paper)

❑Formal specification of the full voting process

❑Non-web based voting interface

❑Analysis of the remote voting protocol

❑Risk and cost analysis of the system

❑Trustworthy voting system

❑Verification of non-Java source code e.g. configuration files

❑Support for other voting systems e.g. US, UK…

❑BON (Business Object Notation) model for KOA

❑MIDP-based mobile voting

❑Formally specifying randomness properties of a system e.g. ballot paper shuffling for anonymity and for use in proportional representation without fractional transfers

❑Incorporating the use of proof carrying certificates (PCC) using the Mobius program verification environment (PVE)

❑Formally Verified Verifiable Voting



Joe Kiniry


Dermot Cochran