e-voting

Seminar Advanced Topics in Cryptography

Selin Sezer

e-voting

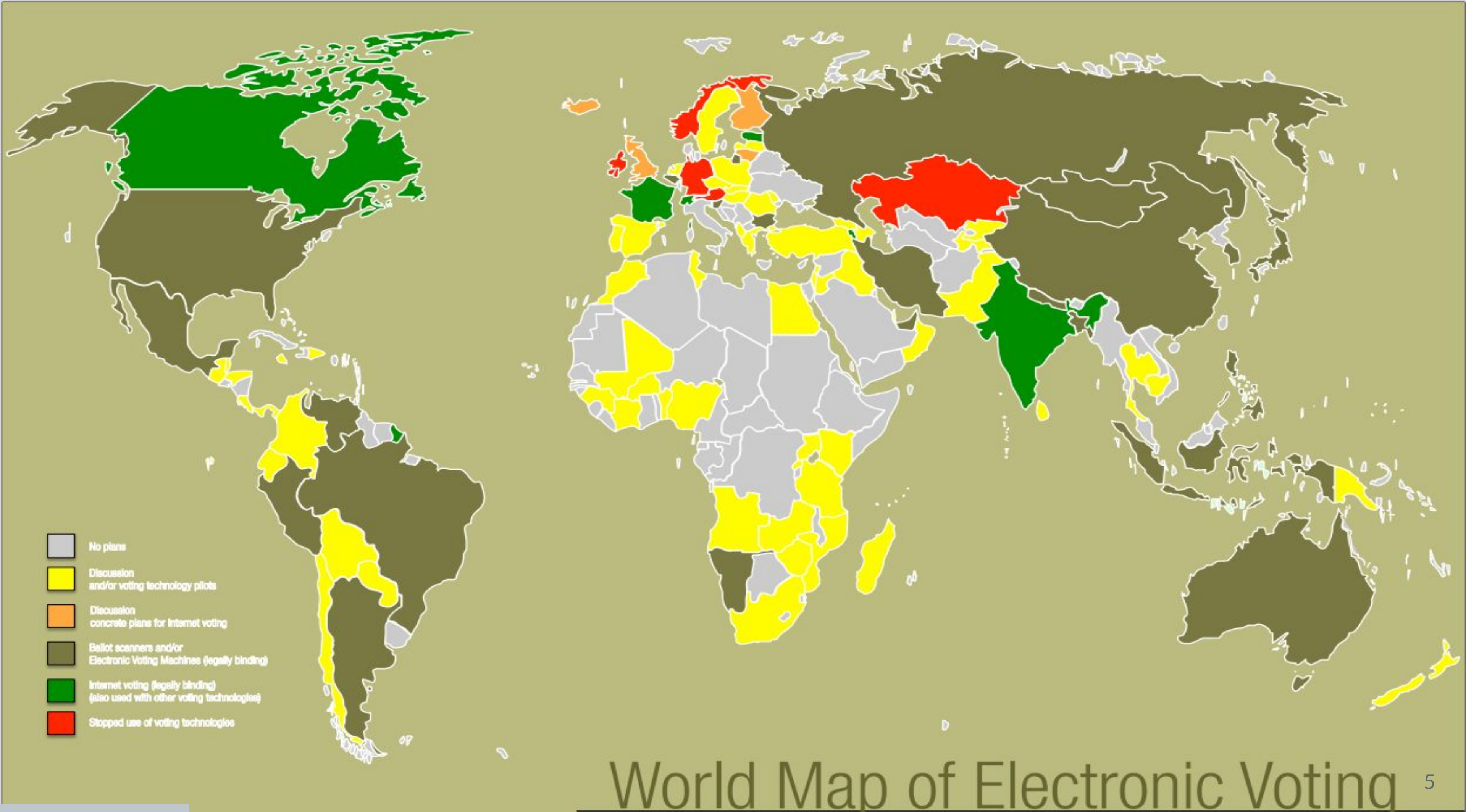
- Introduction
- Authentication in Electronic Elections
- Security Aspects
- Bullet Points From Paper
  - Authentication with Weaker Trust Assumptions for Voting Systems(Quaglia & Smyth)

Introduction

e-voting

- Decision making process

- Hard, conflicting security requirements for remote voting:
  - Integrity
  - Confidentiality


No plans
Discussion
and/or voting technology pilots
Discussion
concrete plans for Internet voting
Ballot scanners and/or
Electronic Voting Machines (legally binding)
Internet voting (legally binding)
(also used with other voting technologies)
Stopped use of voting technologies
World Map of Electronic Voting

Authentication in Electronic Elections

External Authentication*

*Helios(via Facebook, Google), Yahoo(via OAuth)

Γ_{Ext} = (Setup, Vote, Tally, Verify)

Identities: Tallier (T), Voter (V)

<u>T:</u> (pk, sk, mb, mc) $\leftarrow$ Setup(κ)

<u>V:</u> b or ⊥ $\leftarrow$ Vote(pk, nc, v, κ)

<u>T:</u> (V, pf) $\leftarrow$ Tally(sk, nc, bb, κ)

<u>-:</u> s $\leftarrow$ Verify(pk, nc, bb, V, pf, κ)

Internal Authentication**

** Voting system by Juels, Catalano & Jakobsson via cryptographic primitives

Γ_{Int} = (Setup, ***Register***, Vote, Tally, Verify)

Identities: Tallier (T), ***Registrar (R)***, Voter (V)

<u>T:</u> (pk, sk, mb, mc) $\leftarrow$ Setup(κ)

<u>R:</u> ***(pd, d)*** $\leftarrow$ ***Register(pk, κ)***

<u>V:</u> b $\leftarrow$ Vote(***d***, pk, nc, v, κ)

<u>T:</u> (V, pf) $\leftarrow$ Tally(sk, nc, bb, ***L***, κ)

<u>-:</u> s $\leftarrow$ Verify(pk, nc, bb, ***L***, V, pf, κ)

K: security parameter,
pk: public key of tallier,
sk: secret key of tallier,
mb: max. #ballots,
mc: max. #candidates,
pd: public credential,
d: private credential,
nc: some #candidates,
v: voter's vote,
b: ballot,
bb: bulletin board,
L: electoral roll,
pf: non-interactive proof,
V: election outcome vector,
s: election successful bit ∈ {0,1}

Correctness

$(pk, sk, mb, mc) \leftarrow Setup(\kappa)$

for $1 \leq i \leq nb$ do

$\quad$ ***(pd$_i$, d$_i$) ← Register(pk, κ)***

$\quad b_i \leftarrow Vote(\langle \mathbf{d_i} \rangle, pk, nc, v_i, \kappa)$

$\quad V[v_i] \leftarrow V[v_i] + 1$

$(V', pf) \leftarrow Tally(sk, nc, \{b_1, \ldots, b_{nb}\}, \mathbf{\langle\{pd_1, \ldots, pd_{nb}\}\rangle}, \kappa)$

$prob(V = V' \mid nb \leq mb \wedge nc \leq mc) > 1 - negl(\kappa)$

Security Aspects

Smyth, Frink
& Clarkson

Security Aspects

Security Aspects

- Ballot secrecy
- Election verifiability
    - Individual verifiability
    - Universal verifiability
- Eligibility verifiability

Security Aspects

- Ballot secrecy
- Election verifiability
  - Individual verifiability
  - Universal verifiability
- Eligibility verifiability

External Ballot Secrecy Game $G^{\text{Bal-Sec-Ext}}$

- Run election setup (pk, sk, mb, mc) ← Setup(κ).
- Call the attacker A with input 1^{κ} and pk. Await a number nc.
- Set B ← Ø.
- Choose a hidden bit h ← {0,1} randomly.
- Prepare a secrecy oracle O^{Sec}. When called with $v_0, v_1 \in \{1,\dots,nc\}$, the oracle creates ballot b ← Vote(pk, nc, v_h, κ), adds it to B ← B ∪ $\{(b, v_0, v_1)\}$ and returns b.
- Call the attacker A with O^{Sec}. Await a bb.
- Run tally (V, pf) ← Tally(sk, nc, bb, κ).
- Call the attacker *A* with input V and pf. Await a guess h' ∈ {0,1}.
- If $h = h' \wedge \mathit{balanced}(bb, nc, B) \wedge 1 \leq nc \leq mc \wedge //bb// \leq mb$ then ACCEPT
  else REJECT.

> *balanced(bb, nc, B):
> $\forall v \in \{1,\dots,nc\}$ we have
> $|\{b \mid b \in bb \wedge \exists v_1.\ (b,v,v_1) \in B\}| = |\{b \mid b \in bb \wedge \exists v_0.\ (b,v_0,v) \in B\}|$

Definition

An electronic election scheme with external auth.
Γ_{Ext} = (Setup, Vote, Tally, Verify)
satisfies Ballot-Secrecy-Ext
iff
for each ppt attacker *A* the advantage

$$adv^{\text{Bal-Sec-Ext}}(A) = |prob\,(G^{\text{Bal-Sec-Ext}}(A) = \text{ACCEPT}) - ½|$$

is at most *negl(κ)*.

Internal Ballot Secrecy Game $G^{\text{Bal-Sec-Int}}$

- Run election setup (pk, sk, mb, mc) ← Setup(κ).
- ***Call the attacker A with input 1^κ and pk. Await a number nv.***
- ***for 1 ≤ i ≤ nv do***
  $(pd_i, d_i) \leftarrow$ Register(pk, κ).
- Call the attacker A with input ***$\{pd_1,\ldots,pd_{nv}\}$.*** Await a number nc.
- Set B ← Ø, ***R ← Ø.***
- Choose a hidden bit h ← {0,1} randomly.
- ***Prepare a secrecy oracle O^{Sec}. When called with i, adds i to R and returns d_i if i ∉ R. When called with i ∉ R and $v_0, v_1 \in \{1,\ldots,nc\}$, the oracle creates ballot b ← Vote(d_i, pk, nc, v_h, κ) and adds it to B ← B ∪ $\{(b,v_0,v_1)\}$, adds i to R and returns b.***
- Call the attacker A with O^{Sec}. Await a bb.
- Run tally (V, pf) ← Tally(sk, nc, bb, ***$\{pd_1,\ldots,pd_{nv}\}$***, κ).
- Call the attacker *A* with input V and pf. Await a guess h’ ∈ {0,1}.
- If $h = h' \wedge balanced(bb, nc, B) \wedge 1 \leq nc \leq mc \wedge \| bb \| \leq mb$ then ACCEPT
  else REJECT.

Definition

An electronic election scheme with internal auth.
Γ_{Int} = (Setup, ***Register***, Vote, Tally, Verify)
satisfies Ballot-Secrecy-Int
iff
for each ppt attacker *A* the advantage

$$adv^{\text{Bal-Sec-Int}}(A) = |prob\,(G^{\text{Bal-Sec-Int}}(A) = \text{ACCEPT}) - \tfrac{1}{2}|$$

is at most *negl(κ)*.

Security Aspects

- Ballot secrecy
- Election verifiability
  - Individual verifiability
  - Universal verifiability
- Eligibility verifiability

External Individual Verifiability Game $G^{\text{IV-Ext}}$

- Call the attacker A with input 1^{κ}. Await pk, nc, v, v'.
- Run vote algorithm for v and v':
  $b \leftarrow \text{Vote}(pk, nc, v, \kappa)$
  $b' \leftarrow \text{Vote}(pk, nc, v', \kappa)$
- If $b = b' \wedge b \neq \bot \wedge b' \neq \bot$ then ACCEPT
  else REJECT.

Definition

An electronic election scheme with external auth.
Γ_{Ext} = (Setup, Vote, Tally, Verify)
satisfies IV-Ext
iff
for each ppt attacker *A* the advantage

$$\text{adv}^{\text{IV-Ext}}(A) = |\text{prob}\,(G^{\text{IV-Ext}}(A) = \text{ACCEPT})|$$

is at most *negl(κ)*.

Internal Individual Verifiability Game $G^{\text{IV-Int}}$

- Call the attacker A with input 1^κ. Await ***pk and nv***.
- ***for*** $1 \le i \le nv$ ***do***
  (pd_i, d_i) ← ***Register(pk, κ).***
- ***Let*** $L \leftarrow \{pd_1,\ldots,pd_{nv}\}$ ***and Crypt*** $\leftarrow \emptyset$.
- ***Prepare oracle*** O^{IV}. ***When called with*** $i \in \{1,\ldots,nv\}$, ***adds*** d_i ***to Crypt and returns*** d_i.
- ***Call the attacker A with L and*** O^{IV}. ***Await nc, v, v', i, j.***
- Run vote algorithm for v and v':
  b ← Vote(d_i ,pk, nc, v, κ)
  b' ←Vote(d_j ,pk, nc, v', κ)
- If $b = b' \wedge b \neq \perp \wedge b' \neq \perp \wedge i \neq j \wedge d_i \notin Crypt \wedge d_j \notin Crypt$
  then ACCEPT else REJECT.

Definition

An electronic election scheme with internal auth.
Γ_{Int} = (Setup, ***Register***, Vote, Tally, Verify)
satisfies IV-Int
iff
for each ppt attacker *A* the advantage

$$adv^{\text{IV-Int}}(A) = |prob\ (G^{\text{IV-Int}}(A) = \text{ACCEPT})|$$

is at most *negl(κ)*.

Security Aspects

- Ballot secrecy
- Election verifiability
  - Individual verifiability
  - Universal verifiability
- Eligibility verifiability

Algorithm Verify is required to accept iff *the election outcome is correct*.

- The outcome vector length must be nc .
- Component β of Tally outcome vector equals ℓ iff there exist ℓ unique ballots on the bulletin board that are votes for candidate β.
- The output represents the choices used to construct the recorded ballots.

Algorithm Verify is required to accept iff *the election outcome is correct*.

Injectivity

Ballots interpreted only for one candidate.

$(v \neq v' \Rightarrow b \neq b')$

Algorithm Verify is required to accept iff *the election outcome is correct*.

Injectivity

Ballots interpreted only for one candidate.

Completeness

Tally produces election outcomes that will be accepted by **Verify.**

(pr[|bb| ≤ mb ∧ nc ≤ mc => Verify()=1] > 1-negl())

Algorithm Verify is required to accept iff *the election outcome is correct*.

Injectivity	Completeness	Soundness
Ballots interpreted only for one candidate.	**Tally** produces election outcomes that will be accepted by **Verify.**	The probability to conduct a scenario where **Verify** accepts although the election outcome is not correct is negligible. Pr[V*≠V => Verify(V*) = 1]≤ negl()

$\Gamma_{Ext/Int}$ = (Setup, <Register>, Vote, Tally, Verify) satisfies
Universal Verifiability *(UV-Ext/Int)*

if

Injectivity, *Completeness* and *Soundness* are satisfied.

Security Aspects

- Ballot secrecy
- Election verifiability
  - Individual verifiability
  - Universal verifiability
- Eligibility verifiability

Eligibility Verifiability Game $G^{EV\text{-}Int}$

- Call the attacker A with input 1^κ. Await pk and nv.
- for $1 \leq i \leq nv$ do
  $(pd_i, d_i) \leftarrow$ Register(pk, κ).
- Let $L \leftarrow \{pd_1,\ldots,pd_{nv}\}$, $Crpt \leftarrow \emptyset$, and $RvId \leftarrow \emptyset$.
- Prepare oracle O^{EV}. When called with i, v, nc; computes $b \leftarrow Vote(d_i, pk, nc, v, \kappa)$, adds b to RvId and outputs b.
- Prepare oracle O^{IV}. When called with $i \in \{1,\ldots,nv\}$, adds d_i to Crypt and returns d_i.
- Call the attacker A with L, O^{EV} and O^{IV}. Await nc, v, i, b.
- If $b \neq \bot \wedge b \notin RvId \wedge d_i \notin Crpt \wedge \exists r: b = Vote(d_i, pk, nc, v, \kappa\, ; r)$ then ACCEPT else REJECT.

Definition

An electronic election scheme with internal auth.
Γ_{Int} = (Setup, Register, Vote, Tally, Verify)
satisfies EV-Int
iff
for each ppt attacker *A* the advantage

$$adv^{EV\text{-}Int}(A) = |prob\ (G^{EV\text{-}Int}(A) = ACCEPT)|$$

is at most *negl(κ)*.

Authentication with Weaker Trust Assumptions for Voting Systems*

(*)
Elizabeth A. Quaglia and Ben Smyth (2018)
https://eprint.iacr.org/2018/222.pdf

Ext2Int

- $\Gamma_{Ext} \xrightarrow{\text{+digital signature + NIPS}} \Gamma_{Int}$

- Relation R(Γ, Ω) such that
  $((pk, b, \sigma, nc, \kappa), (v, r, d, r')) \in R(\Gamma, \Omega)$
  $\Leftrightarrow$
  $b = Vote(pk, nc, v, \kappa; r) \wedge \sigma = Sign_{\Omega}(d, b; r')$

- $FS(\Sigma, H) = (Prove_{\Sigma}, Verify_{\Sigma})$

- $\Omega = (Gen_{\Omega}, Sign_{\Omega}, Verify_{\Omega})$

- Ext2Int(Γ, Ω, Σ, H) where
  Γ : Underlying election scheme
  Ω : Signature Scheme
  Σ : Sigma Protocol for R
  H : Hash Function

Construction

Ext2Int(Γ, Ω, Σ, H) = (Setup, Register, Vote, Tally, Verify) such that:

Setup(κ): $(pk, sk, mb, mc) \leftarrow Setup_{\Gamma}(\kappa)$

Register(pk, κ): $(pd, (pd,d)) \leftarrow Gen_{\Omega}(pk)$

Vote(d', pk, nc, v, κ): if parse(d') = (pd,d) fails then ⊥ else
pick r, r' at random and compute:
$b \leftarrow Vote_{\Gamma}(pk, nc, v, \kappa; r)$
$\sigma \leftarrow Sign_{\Omega}(d, b; r')$
$\tau \leftarrow Prove_{\Sigma}((pk, b, \sigma, nc, \kappa), (v, r, d, r'), \kappa)$
and outputs (pd, b, σ, τ).

Tally(sk, nc, bb, L, κ): $(V, pf) \leftarrow Tally_{\Gamma}(sk, auth(bb, L), nc, \kappa)$

Verify(pk, nc, bb, L, V, pf, κ): $s \leftarrow Verify_{\Gamma}(pk, auth(bb, L), nc, V, pf, \kappa)$

Ext2Int

auth(bb, L) =

{b | (pd, b, σ, τ) ∈ bb ∧
$Verify_{\Omega}$(pd, b, σ) = 1 ∧
$Verify_{\Sigma}$((pk, b, nc, κ), τ, κ) = 1 ∧
pd ∈ L ∧
(pd, b', σ', τ') ∉ bb \{(pd, b, σ, τ)} ∧
$Verify_{\Omega}$(pd, b', σ') = 1}.

OR

One cannot vote more than once. (Vote once or never)

Ω : Signature Scheme
Σ : Sigma Protocol for R
H : Hash Function

Construction

Ext2Int(Γ, Ω, Σ, H) = (Setup, Register, Vote, Tally, Verify) such that:

Setup(κ): (pk, sk, mb, mc) ← $Setup_{\Gamma}$(κ)

Register(pk, κ): (pd, (pd,d)) ← Gen_{Ω}(pk)

Vote(d', pk, nc, v, κ): if parse(d') = (pd,d) fails then ⊥ else
pick r, r' at random computes:
b ← $Vote_{\Gamma}$(pk, nc, v, κ; r)
σ ← $Sign_{\Omega}$(d, b; r')
$Prove_{\Sigma}$((pk, b, σ, nc, κ), (v, r, d, r'), κ)
b, σ, τ).

Tally(sk, nc, bb, L, v, pf, κ): (v, pf) ← $Tally_{\Gamma}$(sk, auth(bb, L), nc, κ)

Verify(pk, nc, bb, L, v, pf, κ): s ← $Verify_{\Gamma}$(pk, auth(bb, L), nc, v, pf, κ)

Lemma:

Let Γ be an election scheme with external authentication, Ω be a digital signature scheme, Σ be a sigma protocol for relation R(Γ, Ω), and H be a random oracle.

If

Ω satisfies strong unforgeability,

then

Ext2Int(Γ, Ω, Σ, H) is an election scheme with internal authentication.

Security of Ext2Int

- Let Γ be an election scheme with external authentication, Ω be a digital signature scheme, Σ be a sigma protocol for relation R(Γ, Ω), and H be a random oracle.

  If
  Γ satisfies *Ballot-Secrecy-Ext*, Σ satisfies *special soundness* and *special honest verifier zero-knowledge*, and Ω satisfies *strong unforgeability*
  Then
  Election scheme with internal authentication Ext2Int(Γ, Ω, Σ, H) satisfies Ballot-Secrecy-Int.

  Pf. Sketch: ...

Security of Ext2Int

- Let Γ be an election scheme with external authentication, Ω be a digital signature scheme, Σ be a sigma protocol for relation R(Γ, Ω), and H be a random oracle.

  If
  Ω satisfies *strong unforgeability*, Σ satisfies *special soundness* and *special honest verifier zero-knowledge*, and Γ satisfies *UV-Ext*
  Then
  Election scheme with internal authentication Ext2Int(Γ; Ω; Σ; H) satisfies IV-Int, EV-Int, and UV-Int.

  Pf. Sketch: ...

Q&A