پروژه‌ی کارشناسی

اساتید راهنما

دکتر محمد حسین یکتایی

دکتر احسان ملکیان

رای‌گیری الکترونیکی

محسن ملانوری شمس

vip_on_the_web@yahoo.com

محسن مؤمنی

mohsenmomeni@yahoo.com

گروه مهندسی کامپیوتر

دانشگاه تربیت‌معلم تهران

کرج، ایران

شهریور ۱۳۸٤

فهرست مطالب

فصل اول مشخصه‌ی نیازمندی‌های نرم‌افزار

۱-مقدمه

۱-۱-هدف

۱-۲-قواعد به کار رفته در این نوشته

۱-۳-مخاطبان این نوشته

۱-٤-محدوده پروژه

۲-توضیحات کلی

۲-۱-چشم انداز محصول

۲-۲-توابع محصول

۲-۳-طبقه بندی کاربران

۲-٤-محیط عملیاتی

۲-۵-محدودیت طراحی و پیاده سازی

۲-٦-مستندات کاربران

۲-۷-مفروضات و وابستگی ها

۳-نیازمندی های واسط های خارجی

۳-۱-واسط کاربر

۳-۱-۱-واسط کاربر در سمت مشتری

۳-۱-۲-واسط کاربر در سمت سرویس دهنده

۳-۲-واسط سخت افزاری

۳-۳-واسط نرم افزاری

۳-٤-واسط ارتباتی

٤-دیدگاه های سیستم

٤-۱-سیستم از دید رای دهنده

٤-۱-۱-توضیحات و اولویت ها

٤-١-٢-دنباله عمل/عکس العمل
٤-١-٣-نیازمندی های تابعی
٤-٢-سیستم از دیدگاه مدیر سیستم
٤-٢-١-توضیحات و اولویت ها
٤-٢-٢-دنباله عمل/عکس العمل
٤-٢-٣-نیازمندی های تابعی

٥-نیازمندی های غیر تابعی دیگر

٥-١-نیازمندی های کارایی
٥-٢-نیازمندی های ایمنی
٥-٣-نیازمندی های امنیتی
٥-٤-ویژگی های کیفیت نرم افزار

٦-نیازمندی های دیگر

پیوست ١- واژه نامه
پیوست ٢- مدلهای تحلیل

فصل دوم طراحی معماری، واسط و سطح مؤلفه‌ی نرم‌افزار

١) طراحی معماری و ساختار پیمانه‌ای
٢) گزارش پیمانه‌ها
٣) توضیح واسط‌ها
٤) ساختمان‌داده‌های محلی و سراسری
٥) طراحی در سطح مؤلفه

فصل سوم نمودارهای مراحل تحلیل وطراحی

System Arch.(١
CFD(٢
DFD(٣

٤)STD

٥)ERD

فصل چهارم امنیت در سیستم رای‌گیری الکترونیکی

۱)مقدمه

۲)زمینه‌ی تاریخی

۳)مشکلات ذاتی هر نوع رای‌گیری

٤)مشکلات ناشی از بستر انتخاب شده برای سیستم

٥)اشکالات طراحی

٦)دلایل موافقین

۷)نیازهای کلی یک سیستم رای‌گیری الکترونیکی

۸) نتیجه‌گیری

فصل پنجم پیوست‌ها

۱) ترجمه‌ی طرح اولیه‌ی دولت ایرلند برای رای‌گیری الکترونیکی

۲) ترجمه‌ی ارزیابی متخصصین امنیت از نرم‌افزار رای‌گیری الکترونیکی SERVE (نرم‌افزار رای‌گیری الکترونیکی در انتخابات امریکا)

۳) کدهای نوشته شده برای نرم‌افزارها

٤)Java docs of Project

فصل اول:

مشخصه نیازمندی‌های نرم‌افزار

برای

سیستم رای‌گیری الکترونیک

۱-مقدمه

۱-۱-هدف:

هدف از انجام این پروژه ایجاد سیستمی است که بوسیله آن بتوان بر روی شبکه اینترنت (و یا شبکه های دیگر مانند LAN یا WAN) عمل رای گیری و شمارش را بصورت خودکار و بدون دخالت دست انجام داد.

این سیستم مشتمل بر دو قسمت مشتری و سرویس دهنده است که رای دهندگان فقط سیستم مشتری را می بینند. همچنین این سیستم قادر به اخذ رای در مورد افراد، اشیاء یا موضوعات مختلف خواهد بود.

۱-۲-قواعد به کار رفته در این نوشته:

در این نوشته هر جا که احساس شده کلمه ای ابهام برانگیز است از معادل انگلیسی آن استفاده شده است.

۱-۳-مخاطبان این نوشته:

مخاطبان این نوشته توسعه دهندگان، مدیران سیستم و مسئولین رای گیری هستند. این نوشته برای رای دهندگان و کاربران عادی نوشته نشده و لزومی ندارد که این دسته از کاربران از این نوشته با خبر شوند.

۱-٤-محدوده پروژه:

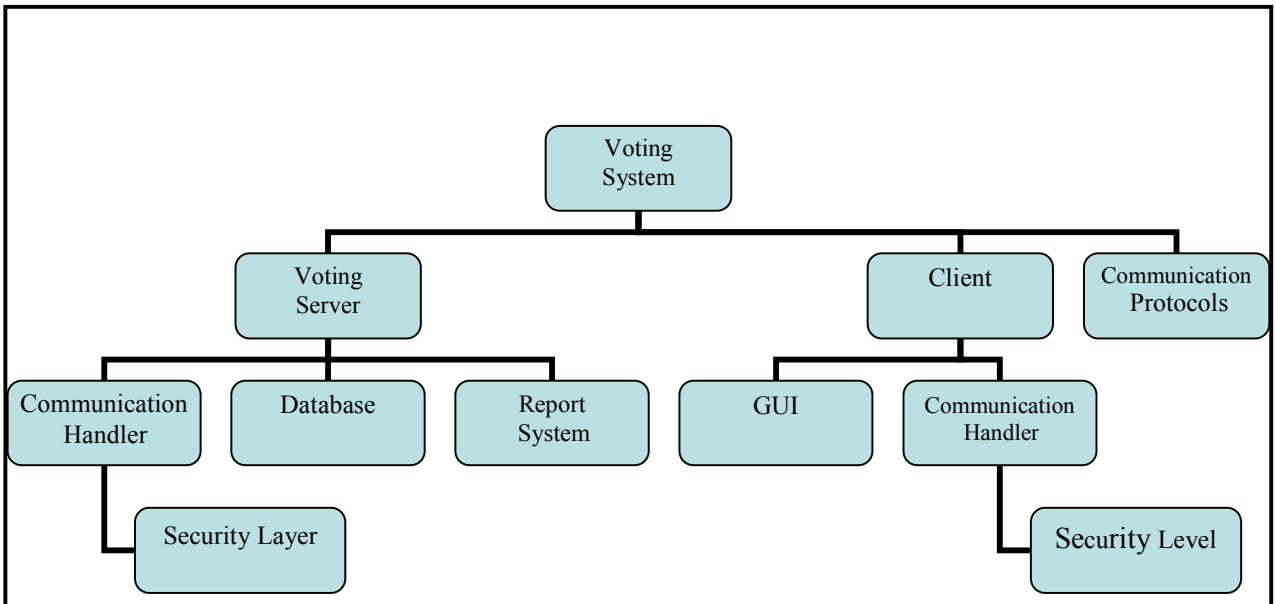
در حال حاظر توسعه این پروژه به منظور اهداف تحقیقاتی انجام می گیرد و استفاده تجاری از آن مد نظر نیست، ولی استفاده از این پروژه در دنیای واقعی نیز دور از انتظار نمی باشد و ممکن است در آینده نزدیک از آن در رای گیری های واقعی نیز استفاده شود.

در دنیای به سرعت در حال توسعه امروز(که در آن اینترنت هر روز نقش بیشتری در زندگی انسانها پیدا می کند) راه گریزی از نهادهای الکترونیکی اجتماعی وجود ندارد. شهرهای اینترنتی، شهروندان اینترنتی و دولت های الکترونیک و نهادهای سیاسی مختلف، همه و همه لزوم ایجاد سیستم های رای گیری الکترونیک را بیش از پیش مسجل می کنند.

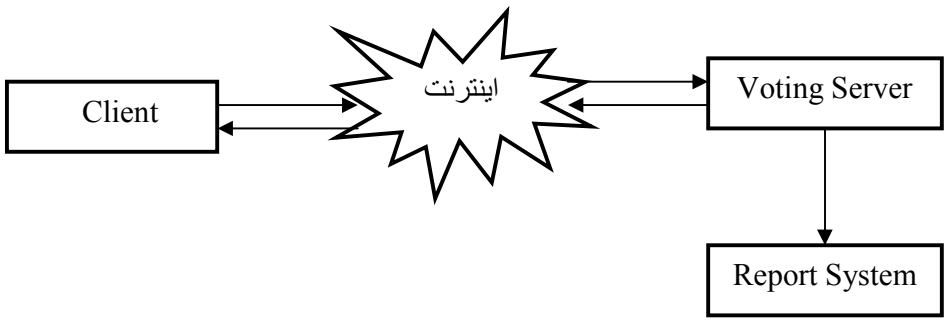
۲-توضیحات کلی:

۲-۱-چشم انداز محصول:

این محصول یک محصول جدید است و ویرایش یا ویرایش هایی از آن قبلا تولید نشده است. شکل زیر نمایی کلی از محصول را نشان می دهد:



شکل ۰-اجزای مختلف سیستم رای گیری الکترونیک



شکل ۱- نمای کلی از سیستم رای گیری الکترونیک

۲-۲-توابع محصول:

محصول تولید شده باید بتواند اعمال زیر را انجام دهد(تشریح این اعمال در بخشهای بعد آمده است):

سرویس دهنده بتواند کاربر را شناسایی و تایید هویت(Authenticate) کند.

نرم افزار مشتری سرویس دهنده را تایید هویت کند.

سرویس دهنده لیست اسامی نامزدها را به مشتری ارائه کند.

کاربر رای خود را انتخاب و ارسال کند.

سرویس دهنده به نحوی مشتری را از رای خود آگاه کند که رای دهنده از اعمال رای خود اطمینان حاصل کند(پیاده سازی این تابع الزامی نیست)

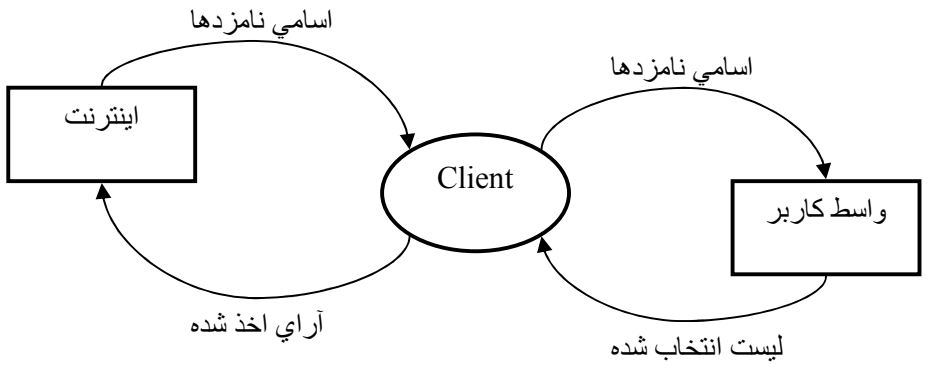
آرای اخذ شده در پایگاه داده ثبت شود.

هر فرد نباید بیش از یک بار رای بدهد.

ارائه گزارش از آرای اخذ شده در طول رای گیری و پس از خاتمه آن.

توانایی تنظیم سیستم برای رای گیری از کاندیدای جدید.

در ادامه نمودار جریان داده از سیستم ایجاد شده را مشاهده می کنید(شکل ۲ و ۳)



شکل ۲- نمودار جریان داده از سیستم رای گیری در سمت مشتری

۲-۳-طبقه بندی کاربران:

کاربران سیستم به دو گروه عمده تقسیم می شوند:

کاربران عادی(رای دهندگان): این دسته از کاربران افراد مختلف اجتماع را تشکیل می دهند، ممکن است افراد تحصیل کرده و کار آزموده و یا افراد آموزش ندیده و مبتدی باشند. در هر حال قسمت عمده کاربران ما در این گروه جای دارند، به همین دلیل باید به این گروه توجه خاصی شود.

مدیران سیستم:افراد آموزش دیده ای هستند که تنظیمات(Setup) و گزارش گیری از سیستم را انجام می دهند.

۲-٤-محیط عملیاتی:

نرم افزار سمت سرور می تواند به یک(یا چند) پلت فرم خاص(مثلا تنها سیستم عامل ویندوز) محدود باشد ولی نرم افزار نوشته شده در سمت مشتری باید بر روی اکثر پلت فرم های متداول موجود قابل اجرا باشد. همچنین نرم افزار سمت مشتری باید با سیستم های موجود سازگاری داشته باشد و استفاده از نرم افزار مشتری نباید به امکانات زیادی نیاز داشته باشد. همچنین نرم افزار مشتری باید تا حد ممکن کوچک باشد(حجم Download کم).

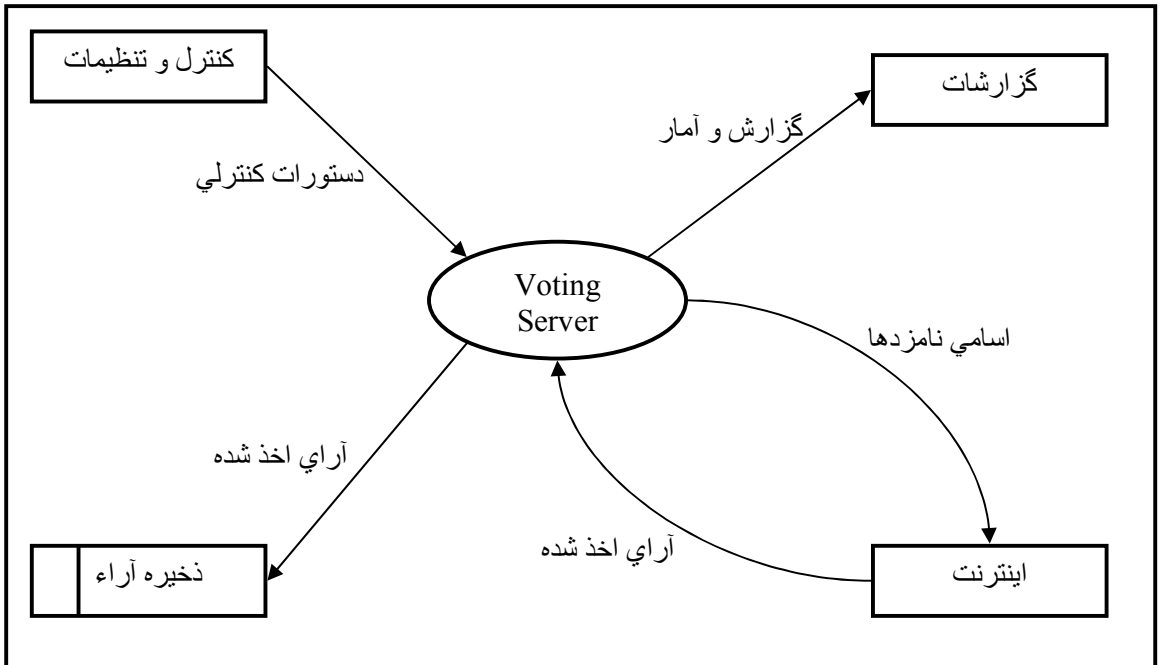
۲-۵-محدودیت طراحی و پیاده سازی:

به علت اهمیت بحث امنیت در این پروژه، نباید از پروتکل های شناخته شده موجود نظیر SSL و HTTPS در انجام این پروژه استفاده شود.

رای دهندگان حتما باید به وسیله کلید خصوصی خود که از مراکز توزیع کلید دریافت می کنند و نه به وسیله اسم کاربری و رمز عبور شناسایی و احراز هویت شوند.

کلید خصوصی باید از طریق یک رسانه قابل حمل(CD، Floppy Disk یا Flash Memory) و نه از طریق شبکه انتقال یابد.

نرم افزار سرور نباید به نرم افزار نوشته شده سمت مشتری اعتماد کند، در صورت بروز هر گونه خطا باید ارتباط قطع و رای باطل شود.



شکل ۳- نمودار جریان داده از سیستم رای گیری در سمت سرویس دهنده

۲-٦-مستندات کاربران:

نرم افزار مشتری باید شامل راهنمایی باشد که کاربر در عرض یک یا دو دقیقه آن را مطالعه کرده و قادر به استفاده از سیستم باشد. نرم افزار سرویس دهنده شامل مستندات کاملی برای استفاده مدیران خواهد بود. برای نگهداری و توسعه سیستم در آینده نیز مستندات طراحی و پیاده سازی موجود خواهد بود.

۲-۷-مفروضات و وابستگی ها:

نرم افزار نهایی تولید شده می تواند از نرم افزار ها و کتابخانه های متن باز موجود استفاده کند. همچنین توسعه دهندگان می توانند از هر گونه ابزار تجاری و متن باز برای توسعه استفاده کنند، ولی نرم افزار نهایی نباید به هر گونه محصول تجاری که برای استفاده کننده منجر به پرداخت هزینه شود وابستگی داشته باشد.

در صورت وابستگی محصول نهایی به یک ابزار و یا محصولی که هزینه ای برای استفاده کنندگان در بر دارد باید معادلی رایگان برای این وابستگی وجود داشته باشد. به عبارت دیگر محصول نهایی باید بتواند به نحوی بدون پرداخت هزینه اضافی برای نرم افزارهای دیگر مورد استفاده قرار گیرد.

۳-نیازمندی های واسط های خارجی:

۳-۱-واسط کاربر:

واسط کاربر در این پروژه به دو بخش سمت مشتری و سمت سرور تقسیم می شود

۳-۱-۱-واسط کاربر در سمت مشتری:واسط کاربر در سمت مشتری باید در حد امکان ساده باشد و نباید کاربر با نوشته ها، دکمه ها و اشکال گوناگون گیج شود. واسط سمت مشتری باید امکانات زیر را در اختیار رای دهنده قرار دهد:

کاربر باید بتواند محل کلید خصوصی خود را(که در یک فایل قرار دارد) انتخاب کند و یا مقدار پیش فرض آن را تغییر دهد.

کاربر باید آرای خود را به راحتی انتخاب کند.

واسط باید لیست انتخاب شده را دوباره به کاربر نشان دهد تا او از انتخاب خود اطمینان حاصل کند.

کاربر باید پس از ارسال رای خود از آخرین نتایج با خبر شود.

همچنین واسط کاربر باید اشتباهات کاربر را به گونه ای منطقی به او نشان دهد و یا از انجام اشتباهات جلوگیری کند. برای مثال هنگامی که کاربر امکان انتخاب یک گزینه را دارد، تنها بتواند یک گزینه را انتخاب کند و هنگامی که امکان انتخاب چند گزینه را دارد به او اجازه انتخاب چند گزینه داده شود.
همچنین توصیه می شود مراحل انجام کار بصورت مرحله به مرحله(Wizard) به کاربر نشان داده شود.
نکته مهم دیگر در مورد واسط کاربر در سمت مشتری این است که واسط باید توانایی نشان دادن اسم نامزدها به همراه عکس آن ها و مشخصات دیگری از آنها را داشته باشد.

۳-۱-۲-واسط کاربر در سمت سرویس دهنده:واسط کاربر در سمت سرویس دهنده باید به صورت خط فرمانی و نه به صورت گرافیکی باشد(در بسیاری از سرویس دهنده ها محیط گرافیکی وجود ندارد). این واسط باید توانایی تعریف نامزدهای انتخاباتی و تعیین مشخصات آن ها نظیر نام، سوابق و عکس آن ها را فراهم کند. همچنین واسط باید گزارشات لازم از قبیل تعداد کل آرای اخذ شده و درصد و تعداد آرای اخذ شده توسط هر نامزد را به شکلی مناسب و قابل درک(مثلا فایل HTML) در اختیار مدیران قرار دهد.

۳-۲-واسط سخت افزاری:
حداقل سخت افزار مورد نیاز در سمت سرور یک دستگاه PC با پردازنده Pentium 4 و ۵۱۲ مگا بایت حافظه به علاوه سخت افزار لازم برای اتصال به شبکه(کارت شبکه و ...) است. البته ممکن است در پلت فرم هایی غیر از ویندوز از سخت افزارهای دیگری استفاده شود. همچنین تمام گرداننده(Driver)های لازم باید توسط سیستم عامل فراهم شود و توسعه دهندگان ملزم به پیاده سازی هیچ گونه گرداننده ای نمی باشند.
در طرف مشتری نرم افزار باید بر روی یک دستگاه PC با پردازنده Pentium III و ۱۲۸ مگا بایت حافظه و سخت افزار لازم برای اتصال به شبکه قابل اجرا باشد.
۳-۳-واسط نرم افزاری:
در سمت سرور نرم افزار برای اجرا به محیط اجرای جاوا(JRE) ویرایش ۱,۵ یا بیشتر نیاز دارد. قابل ذکر است که JRE بر روی بسیاری از سیستم عامل ها از جمله Windows, Linux, Mac OS و Solaris قابل نصب است.

همچنین نرم افزار سمت سرور برای اجـرا بـه یکـی از DBMS هـای MS SQL Server 2000، MySql ، Oracle و یا HSQL DB نیاز خواهد داشت.
در سمت مشتری نرم افزار باید بر روی JRE ویرایش ۱,٤ یا بیشتر اجرا شود.

۳-٤-واسط ارتباتی:
نرم افزار در دو سمت مشتری و سرویس دهنده به پروتکـل هـای لایـه چهـارم نظیـر HTTP نیـازی ندارد(گر چه این پروتکل تقریبا بوسیله همه سیستم عامل ها پشتیبانی می شود). نرم افزار برای اجرا به پشتیبانی سیستم عامل از پروتکل TCP نیاز دارد. همچنین برای مشاهده گزارشات به یک مرورگر وب نیاز است.
نرم افزار برای رمزنگاری از روش AES و کلیدهایی با استاندارد X.509 استفاده می کند.
همچنین خطوط ارتباطی باید پهنای باند لازم برای اتصال هم زمان چندین مشتری را فراهم کنند.

٤-دیدگاه های سیستم
٤-۱-سیستم از دید رای دهنده:
٤-۱-۱-توضیحات و اولویت ها:رای دهندگان مهمترین نقش را در سیستم بـازی مـی کننـد، از سـوی دیگر عموم آنها افرادی هستند که برای اجرای نقش خود آموزش ندیده اند و شاید حتـی آشـنایی لازم را با کامپیوتر را نیز ندارند. به همین دلیل این گروه از کاربران از اهمیت و اولویت بالایی برخوردارند.
٤-۱-۲-دنباله عمل/عکس العمل: دنباله عمل/عکس العمل از دیدگاه رای دهنده به قرار زیر است:
رای دهنده برنامه را شروع می کند ← برنامه شروع به کار کرده و محـل کلیـد خـصوصی را از کـاربر تقاضا می کند.
کاربر محل کلید خصوصی را(که بصورت یک فایل است) مشخص می کند ← به ترتیب مراحل زیـر انجام می شود:
برنامه با سرویس دهنده ارتباط بر قرار می کند
مشتری، سرویس دهنده را احراز هویت می کند(تا مطمئن شود سرویس دهنده واقعی است)
سرویس دهنده، مشتری را احراز هویت می کند(تا ببیند چه کسی رای می دهد)
لیست اسامی نامزدها به برنامه مشتری ارسال می شود
لیست اسامی به رای دهنده نشان داده می شود.

رای دهنده نامزدهای مورد علاقه خود را انتخاب می کند ← برنامه یک بار دیگر نامزدهای انتخاب شده را به کاربر نشان می دهد و از او تایید می گیرد.

رای دهنده اسامی نامزدهای انتخابی را تایید می کند ← مراحل زیر به ترتیب انجام می شود:

برنامه لیست انتخاب شده را به سرور می فرستد

سرور لیست دریافت شده را در پایگاه داده ثبت می کند

سرور آماری از آخرین آرای انتخابی را از پایگاه داده گرفته و به سمت مشتری می فرستد

لیست دریافتی در سمت مشتری به رای دهنده نشان داده می شود

رای دهنده کلید خروج را فشار می دهد ← برنامه به کار خود پایان می دهد

٤-١-٣-نیازمندی های تابعی:جزئیات نیازمندی های تابعی برای نرم افزار مشتری در ادامه آمده است:

واسط کاربر برای انتخاب کلید خصوصی باید یک Brows File Dialog از نوعی که در سیستم عامل مربوطه متداول است به او نشان دهد و فایل انتخاب شده را دریافت کند.

در صورتی که فایل انتخاب شده، صحیح نبود باید به او پیغام خطایی مبنی بر اشتباه بودن انتخاب وی نشان داده شود و به کاربر اجازه انتخاب مجدد داده شود.

کاربر در هر مرحله از اجرای برنامه و قبل از ارسال آرا باید بتواند از رای دادن انصراف بدهد.

کاربر در هنگام برقراری ارتباط و رد و بدل شدن اسامی باید از مراحل انجام کار با خبر شود.

در صورتی که در هنگام برقراری ارتباط مشکلی پیش آمد باید اشکال به زبانی ساده به کاربر نشان داده شده و به او شانس سعی مجدد داده شود.

در صورتی که احراز هویت سرور به درستی انجام نشد، باید به کاربر اطلاع داده شده و از ادامه کار وی جلوگیری به عمل آید(خروج از برنامه)

شکل برنامه باید به شکلی باشد که کاربر تعداد درستی از کاندیداها را انتخاب کند و در صورتی که تعداد نادرستی از کاندیداها انتخاب شده بودند به کاربر اجازه ارسال رای داده نشود.

پس از انتخاب اسامی، باید این اسامی دوباره به کاربر نشان داده شده و از او تایید گرفت. در این مرحله کاربر باید بتواند رای خود را مجددا تغییر دهد.

پس از ارسال رای و دریافت آمار آرا، این آمار باید به شکل نمودار، به همراه تعداد آرا و درصد آنها به کاربر نشان داده شود.

پس از خاتمه یک مرحله از رای دهی نیازی نیست که کاربر بتواند مجددا رای بدهد و برای انجام این کار می تواند مجددا برنامه را اجرا کند.

٤-٢-سیستم از دیدگاه مدیر سیستم:

٤-٢-١-توضیحات و اولویت ها:استفاده مدیران از سیستم به تعداد دفعات کمتری صورت می گیرد، همچنین عموما مدیران سیستم افرادی کار آزموده و فنی هستند. به همین دلیل سادگی کار با سیستم مدیریت اولویت کمتری نسبت به سیستم مشتری دارد.

گر چه سیستم از دیدگاه مدیر از پیچیدگی بیشتری برخوردار است ولی مراحل انجام کار در این طرف کمتر است.

٤-٢-٢-دنباله عمل/عکس العمل:دنباله عمل از دیدگاه مدیر سیستم به قرار زیر است:

مدیر دستور شروع سرور را می دهد ← سرور شروع به کار می کند

مدیر فرمان گزارش گیری را صادر می کند ← گزارش تهیه شده در یک فایل HTML ذخیره می شود.

٤-٢-٣-نیازمندی های تابعی:جزعیات نیازمندی های تابعی از دید مدیر سیستم به قرار زیر است:

در سمت سرویس دهنده نیازی به واسط کاربر گرافیکی نیست و همه تنظیمات بوسیله فایل های متنی انجام می شود. برای انجام تنضیمات دوفایل مختلف، یکی برای تنظیمات سرور و یکی برای اسامی نامزدها وجود دارد.

در فایل اسامی نامزدها، مدیر باید بتواند مشخصات نامزدها شامل نام و نام خانوادگی، توضیحی کوتاه در مورد آنها و عکس نامزدها را تعیین کند. همچنین تعداد نامزدهایی که کاربر می تواند انتخاب کند نیز در این فایل انتخاب می شود.

در فایل تنظیمات سرور مدیر باید قادر به تعیین شماره پورت سرور و محل کلیدهای خصوصی رای دهندگان باشد.

مدیر برای شروع به کار سرور فرمانی را در خط فرمان تایپ می کند

در صورتی که در هنگام شروع به کار سرور اشکالی پیش آمد، باید پیامی به کاربر داده شده و اجرای برنامه خاتمه یابد.

پس از شروع به کار، سرور منتظر ارتباطی از سوی مشتری می شود و تا پایان اجرا به این کار ادامه می دهد.

در صورتی که در حین اجرای سرور مشکلی پیش آمد، سرور نباید به کار خود خاتمه دهد مگر اینکه این مشکل باعث شود که سرور نتواند به هیچ یک از مشتریان خدماتی بدهد.

برای گزارش گیری از سیستم باید فرمانی موجود باشد. با اجرای این فرمان یک فایل HTML که شامل جدولی از آمار آرای داده شده است تولید می شود. این فرمان دارای یک آرگومان ورودی است که محل فایل خروجی را تعیین میکند.

۵-نیازمندی های غیر تابعی دیگر:

۵-۱-نیازمندی های کارایی:

نرم افزار سرور باید بر روی حداقل سخت افزار لازم(ذکر شده در قسمت ۲-۳) قادر به پاسخ دهی همزمان به حداقل ۱۰ مشتری در زمانی کمتر از ۵ ثانیه باشد(دقت شود که تاخیرهای ناشی از شبکه و عکس العمل کاربر نیز باید به این زمان اضافه شود)

۵-۲-نیازمندی های ایمنی:

نرم افزار سرور باید قادر به اجرا در یک پوشه(directory) باشد بدون اینکه به فایل های خارج از آن دسترسی داشته باشد. البته فایل هایی که توسط DBMS تولید می شود می تواند خارج از این پوشه قرار بگیرد.

نرم افزار مشتری فقط باید از فایل کلید خصوصی اطلاعات لازم را بخواند و اجازه نوشتن روی هیچ قسمت از دیسک را ندارد.

۵-۳-نیازمندی های امنیتی:

همان طور که قبلا نیز ذکر شد امنیت از مهمترین اهداف این پروژه است. هر کاربر تا هنگامی که احراز هویت نشده اجازه انجام هیچ کاری را ندارد. احراز هویت سرویس دهنده و مشتری توسط کلید خصوصی آنها با فرمت X.509 انجام می گیرد و برای رمز نگاری از استاندارد AES استفاده می شود. تمام اطلاعات مبادله شده در حالت رمز شده انتقال می یابد.

۵-٤-ویژگی های کیفیت نرم افزار:

پس از امنیت مهمترین خصیصه لازم برای نرم افزار صحت(Correctness) است. نرم افزار نباید در هیچ یک از مراحل اخذ رای و شمارش آرا خطایی داشته باشد. ویژگی مهم دیگر در دسترس بودن(Availability) نرم افزار است. چون ساعات رای گیری محدود است، نرم افزار باید در این

ساعات همواره در دسترس باشد. همچنین نرم افزار باید دیگر ویژگی های لازم یک نرم افزار خوب را تا حد ممکن دارا باشد.

٦-نیازمندی های دیگر:

نرم افزار سمت مشتری باید قابلیت بین المللی شدن را داشته باشد و بتوان زبان جدیدی بـه آن اضـافه کرد بدون این که به کامپایل مجدد نیازی باشد. توجه شود که نیازی نیست که رای دهنده در زمان اجرا بتواند چنین کاری را انجام دهد بلکه تغییر یا اضافه کردن زبان توسط مدیر سیستم انجام می شود.

پیوست ۱- واژه نامه:

رای گیری الکترونیک:e-voting

Java Runtime Environment:JRE

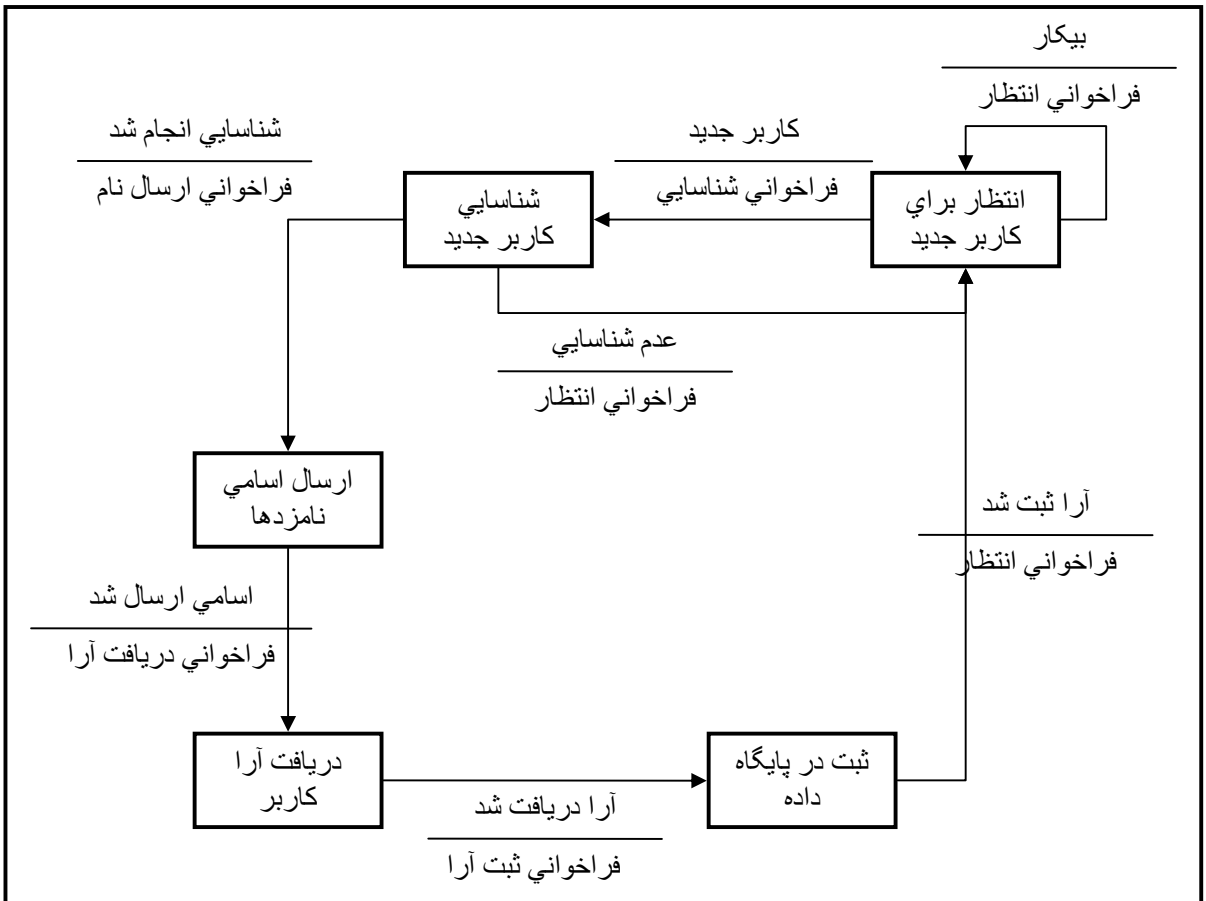
Advanced Encryption Standard:AES – یک روش رمز نگاری به وسیله کلید نامتقارن است.

مشتری:Client

سرویس دهنده:Server

X.509: یک فرمت استاندارد برای نگهداری کلید عمومی و خصوصی و اطلاعات دیگری از کـاربران است.

پیوست ۲- مدل‌های تحلیل:



شکل ٤- نمودار تغییر حالت برای نرم افزار سرویس دهنده

فصل دوم

طراحی معماری ، واسط و سطح مولّفه نرم افزار

رای گیری الکترونیکی

۱) طراحی معماری و ساختار پیمانه‌ای

طراحی معماری

نشاندهنده ساختارداده ها و مولفه های برنامه ای که برای ایجاد سیستم کامپیوتری مورد نیاز است می باشد .

معماری نهائی سیستم

ساختار و خصوصیات مولفه های سیستم

و روابط بین مولفه های معماری سیستم

در طراحی معماری نشان داده می شوند .

مراحل انجام طراحی معماری

۱) نگاشت تبدیلها و تراکنشها

بر اساس آنچه در قسمت تحلیل نیازها انجام شده و با توجه به DFD های موجود نگاشت تبدیلی و تراکنشی را مرحله به مرحله انجام می دهیم.

۲) گزارش پیمانه ها

در این مرحله برای هر یک از پیمانه هایی که در مرحله قبل معرفی شده اند ، گزارشی مختصر در مورد آنچه در آنها انجام می شود ، چگونگی ارتباط آنها و طریقه ارسال داده ها توضیح مختصری داده می شود.

۳) تعریف و توضیح واسط هر پیمانه

در این مرحله برای هر یک از پیمانه ها یک واسط طراحی می شود ، که سرویسهای ارائه شده توسط هر پیمانه را مشخص می نماید.

٤) ساختارهای داده محلی و سراسری

قالب داده ها و فرمت استاندارد ساختارهای داده ای محلی ، سراسری و پیامهای ارسال شده در این قسمت تعیین می شود .

٥) محدودیتهای طراحی معماری

در این قسمت همه محدودیتها (limitations or constraints) یی که اثر مهمی بر طراحی سیستم دارند ذکر می گردند.

این محدودیتها می توانند بوسیله هریک از عناصری که در ادامه می آید ، بوجود آیند :

محیط سخت افزاری و نرم افزاری

محیط کاربر نهایی (end-user)

در دسترس بودن و اعتبار منابع

مطلوبیتها (compliance) استاندارد

نیازهای Interoperablity

نیازهای واسط/پروتکل

نیازهای توزیع و تجمیع داده ای (Data repository and distribution)

نیازهای امنیتی

محدودیتهای حافظه یا گنجایش دیگر تجهیزات

محدودیتهای کارایی

ارتباطات شبکه

نیازهای تست نرم افزار (Verification & Validation)

دیگر وسایل کیفیت آدرس دهی

دیگر نیازهایی که در مشخصه نیازها توصیف شده اند.

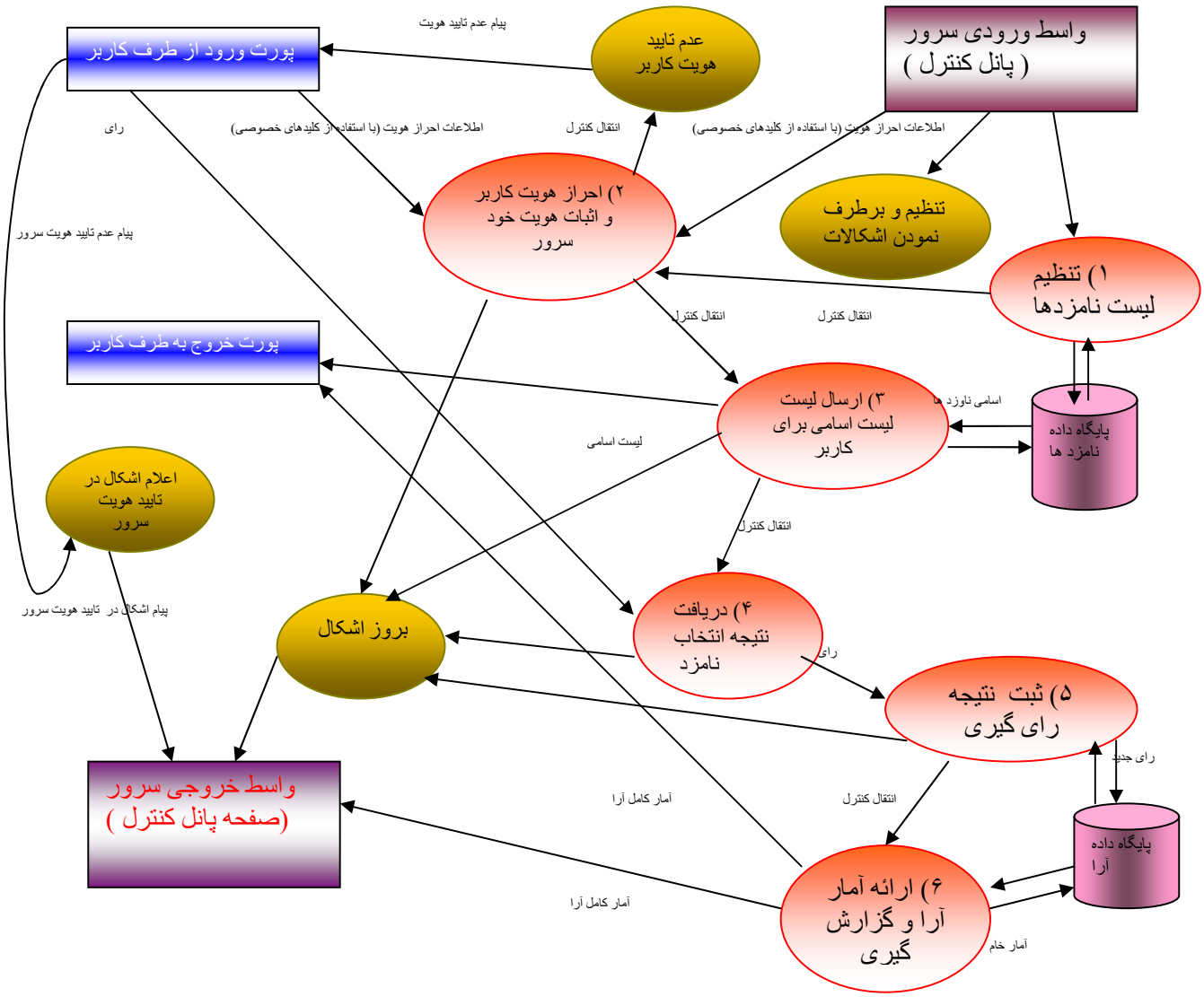
٦) مجموعه بازنگریها و اصلاح

فرآیند طراحی ، یک فرآیند تکراری است ، که در آن هر بار اصلاحات در طراحی قبلی انجام می گیرد . بنابراین شایسته است برای اگاهی از مراحل مختلف ان نمودارها و طراحیها در step های مختلف اورده شوند تا روند تکامل طراحی مشخص گشته از بروز خطاها در مراحل بعدی تولید

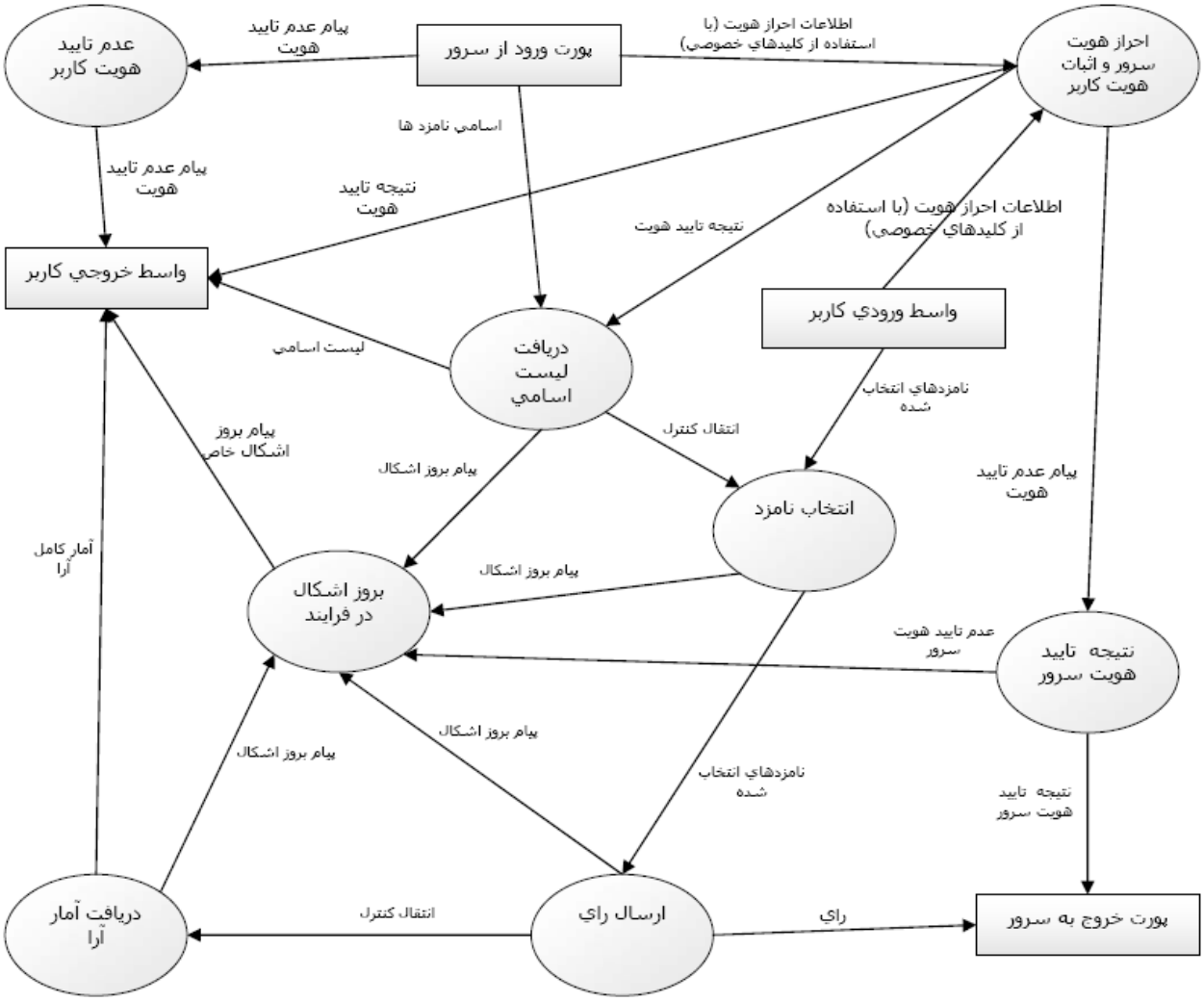
جلوگیری گردد و اشکالات به صورت واضحتری دیده شوند . همچنین باز نگری نهایی برای همخوانی کلیه قسمتهی طراحی امری لازم به نظر می رسد .

معماری در ســـــاده ترین شکل خود عبارت است از ساختار سلسله مراتبی اجزاء برنامه (پیمانه ها)، شیوهٔ ارتباط این اجزاء و ساختار داد ه هایی که توسط اجزاء مورد استفاده قرار می گیرند.

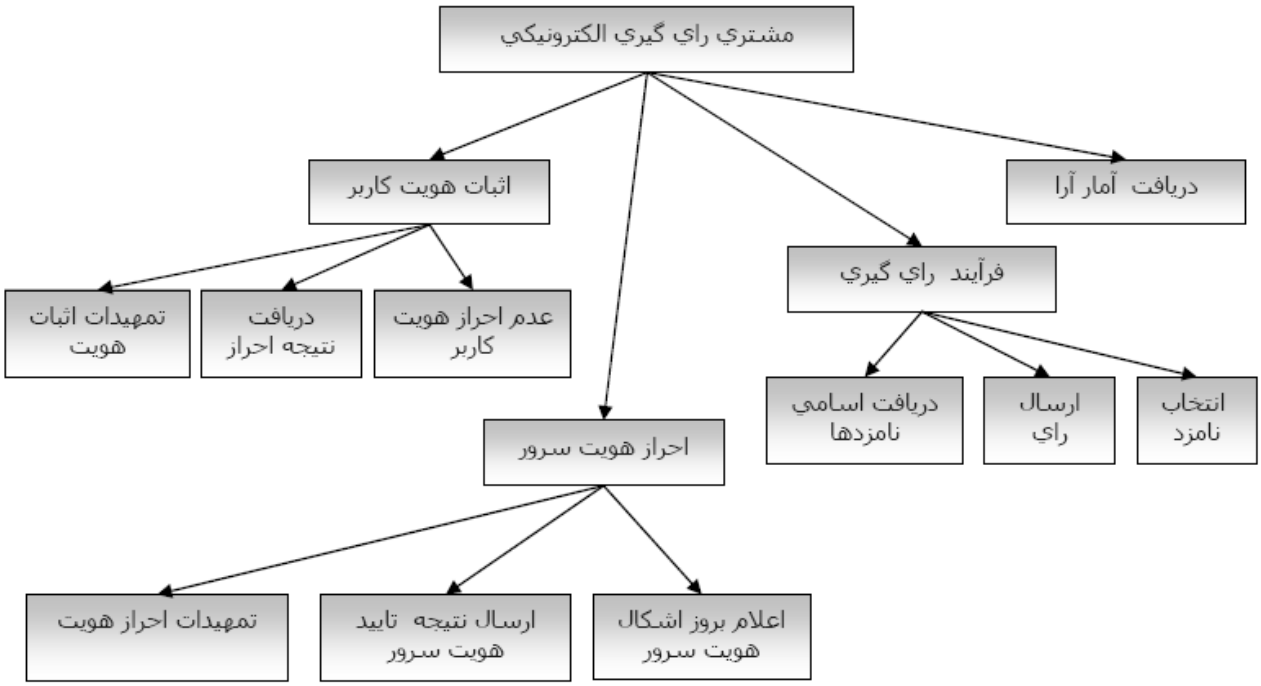
در ایجاد این معماری باید ابتدا معماریهای نرم افزار سمت سرور و مشتری را از هـم تفکیـک نمـود و براین اساس معماری کلی دو نرم افزار را به تفکیک از هم نشان داد . برای این منظور ابتدا باید DFD های مجددی تنظیم نمود که در آنها جریان داده در نرم افزار های سمت سـرور و مـشتری بـه شـکل جداگانه و منفک نشان داده شده باشند و سپس از روی آن طبق مراحل هفـت گانـه نگاشـت تبـدیل و تراکنش را انجام داد .



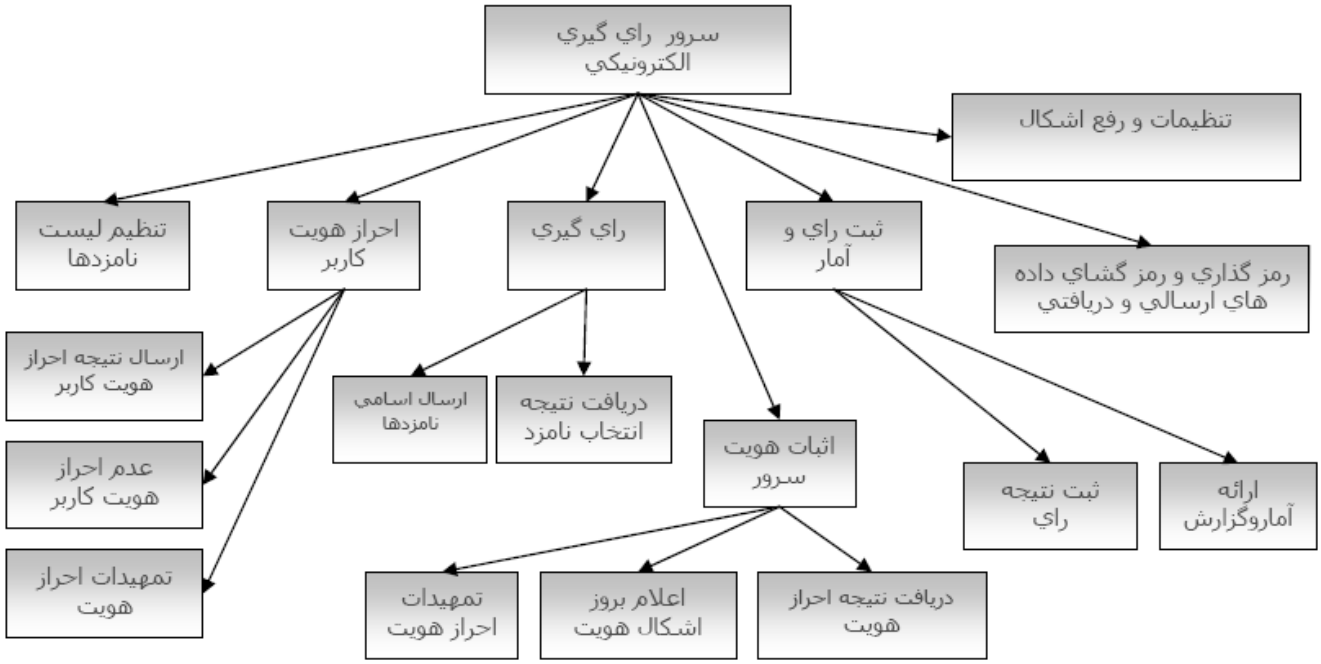
سطح ۱ نرم افزار سرور رای گیری DFDشکل۳-۱



شکل۳-۲ DFD سطح ۱ نرم افزار کاربر راي گیري الکترونیکي



شكل ۳-۳ معماري كلي و اوليه نرم افزارمشتري راي گيري الكترونيكي



شكل ۳-۴ معماري كلي و اوليه نرم افزارسرور راي گيري الكترونيكي

۲) گزارش پیمانه ها

در این مرحله به گزارش آنچه در هر یک از پیمانه ها انجام می شود می پردازیم :

نرم افزار کارگزار رای‌گیری الکترونیکی

تنظیم لیست نامزدها و تنظیمات اولیه

در این پیمانه لیست نامزدها توسط admin سرور ، از طریق واسط پانل کنترل تنظیم و در پایگاه داده ذخیره می گردد . مواردی از قبیل الگوریتم رمز نگاری و احراز هویت نیز در این مرحله توسط Admin تعیین می شود . تا زمانیکه این مرحله انجام نشود سرور برای ارائه خدمات به کاربران آماده نیست.

احراز هویت کاربر

در این مرحله به احراز هویت کاربر پرداخته می شود . و نتیجه برای کاربر فرستاده می شود . چنانچه کاربر از عهده این مرحله برنیاید و هویتش تایید نشود ، سرور ارتباط را به صورت یکطرفه قطع خواهد نمود . سه پیمانه جداگانه مسئول انجام این اعمال هستند ،که پیمانه فوق با احضار آنها کار خود را انجام می دهد .

تمهیدات احراز هویت کاربر

در این مرحله کلیه اعمالی که لازم است برای احراز هویت انجام شود ، از قبیل ارسال رشته تصادفی ، رمز کردن با استفاده از کلید عمومی کاربر و ... انجام می پذیرد . توصیف کامل این مرحله در بخش طراحی مولفه می آید .

ارسال نتیجه احراز هویت کاربر

پس از آنکه احراز هویت انجام شد نتیجه در قالب پیامی که نشاندهنده تایید یا عدم تایید هویت کاربر است به او ارسال می گردد . با استفاده از نتیجه ای که این مرحله به پیمانه احراز

هویت کاربر می فرستد ، تصمیم گیری در مورد انتقال کنترل انجام می شـود ، کـه در صـورت تاییـد انتقال کنترل به مرحله اثبات هویت سرور (با برگرداندن نتیجه درست به پیمانه اصلی) و در صـورت عدم تایید به مرحله عدم تایید هویت کاربر منتقل می شود . نکته مهم این است که پیام ارسال شده در این مرحله قالب استانداردی داشته باشد ، تا نسخه های مختلف مشتری بتوانند از ایـن سـرور اسـتفاده کنند . (یعن نرم افزار مشتری بتوند توسط هر کسی تولید گردد .)

عدم تایید هویت کاربر

چنانچه در مرحله قبل هویت کاربر تایید نگردد ، کنترل به این مرحله منتقل می شود ، که در آن تنها کاری که انجام می شود این است که ارتباط به صورت یکطرفه قطع می گـردد و کنتـرل به واسطه خروجی ارسا از این مرحله به مرحله انتظار برای ارتباط کاربر باز می گردد . این مرحلـه کـه در معماری اولیه لحاظ نشده در قسمت اصلاح اضافه می گردد . و بدین شکل معماری کلی به صورت شکل۳-۵ تغییر می یابد .

اثبات هویت سرور

آنچه در این مرحله انجام می شود دقیقا همان کار مرحله قبل است که در جهت مخـالف صـورت می پذیرد . البته این بار در صورت عدم احراز هویت سرور ،یک مشکل تشخیص داده شده و از طریق صفحه پانل کنترل برای Admin سرور پیغامی ارسال می گـردد کـه نـشانگر بـروز مـشکل در تاییـد صلاحیت است . و توسط Admin اصلاح تنظیمات به شکلی صورت می گیرد که این مشکل مرتفـع شود .

تمهیدات اثبات هویت سرور

در این مرحله همچون آنچه در تمهیدات احراز هویت کـاربر انجـام شـد ولـی اینبـار بـر عکس انجم می شود . رمز گشایی رشته رمز شده توسط کلید خصوصی و دیگر عملیات در این مرحله صورت می پذیرد .

دریافت نتیجه تایید هویت

در این گذر نتیجه تایید هویت در قلب پیامی از طرف کاربر فرستاده شده و توسط سرور دریافت می شود ف این نتیجه مستقیما به پیمانه بالا یعنی پیمانه اثبات هویت سرور فرستاده می شود تا براساس این نتیجه تصمیم گیری در مورد انتقال کنترل صورت پذیرد .

قالب پیغامی که در این مرحله رد و بدل می شود نیز از استاندارد مرحله قبلی تبعیت می کند . این قالب در قسمت ساختمان داده های طراحی تشریح می گردد .

اعلام بروز اشکال هویت

در صورتی که اشکالی در تایید هویت بروز کند ، کنترل به این قسمت منتقل می شود که در آن با واسط Admin ارتباط برقرار شده و بروز مشکل و نوع آن نشان داده می شود . سپس انتقال کنترل به مرحله انتظار برای کاربر جدید انجام می شود .

چند نکته مهم در مورد این طراحی وجود دارد :

اثبات هویت سرور و احراز هویت کاربر که در دو پیمانه جداگانه تفکیک شده است ، معمولا به شکل منفک از هم انجام نمی شود . این دو مرحله یا به صورت همروند با هم انجام می شوند ویا در قالب یک پیمانه با هم ترکیب خواهند شد . ما در طراحی موله ای ین دو پیمانه را در هم ادغام می کنیم و صورت کلی سلسله مراتب پیمانه ها را به شکل زیر در می آوریم:

احراز هویت کاربر و اثبات هویت سرور

تمهیدات احراز و اثبات هویت

ارسال نتیجه تایید هویت

دریافت نتیجه احراز هویت کاربر

عدم تایید هویت کاربر

اعلام بروز اشکال هویت سرور

اما در شکل عملیاتی پیمانه ها تغییری حاصل نمی شود و گزارش عملیات صورت گرفته در انها به همین صورتی که در بالا ذکر شد خواهد بود .

احراز هویت می تواند با استفاده از رمز نگاری کلید متقارن و یا نا متقارن صورت گیـرد کـه اولـی در مواردی کهامنیت رای گیری اهمیتی به ندازه سرعت و کارایی برنامه ندارد و دومی در صورت نیـاز بـه امنیت بالا مورد استفاده قرار می گیرد و تنظیم شکل Authentication در تنظیمـات اولیـه ای کـه توسط Admin صورت می پذیرد انجام می شود که الگوریتم رمز نگاری و احراز هویت را تعیین می کند .

یاد آوری می کنیم که کلیه داده های ارسالی برای کاربر ، پس از این مرحله قبل از ارسال از یک پیمانه رمز گذار عبور می کنند که برای تضمین امنیت سیستم همه داده ها را به صورت رمز شده ارسال مـی کند . این رمز گذاری به صورت متقارن و با استفاده از کلیدی که در مرحله احراز هویت سـاخته مـی شود ، انجام می شود . همچنین از این پس داده های دریافت شده از طرف کاربر نیز رمز شده اسـت و برای رمز گشایی به این پیمانه ارسال می شود . این پیمانه به منظور جلـوگیری از ایجـاد پیچیـدگی و ابهام در DFD آورده نشده است .

تشکیل session ارتباطی با کاربر نیز در ان مرحله صورت می گیرد و کلید session که همـان کلیـد رمز گذاری داده است و تا پایان ارتباط با این کاربر session ایجاد شده حفظ می شود .

معماری کلی نرم افزار در سطح بالا از این پس به شکل ۳-۵ تغییر می کند ، کـه در آن مراحـل ایجـاد session و ... لحاظ شده است .

در صورت بروز هر گونه اشکال در این پیمانه ها و پیمنه های دیگر ، کنترل به پیمانـه بـروز اشـکال منتقل می شود و session ختم شده ، و اعمال لازم در پیمانه بـروز اشـکال صـورت مـی گیـرد و در نهایت این پیمانه، اشکال را برای صفحه پانل کنترل می فرستد و کنترل را به پیمانه انتظار برای کـاربر جدید ، وامی گذارد .

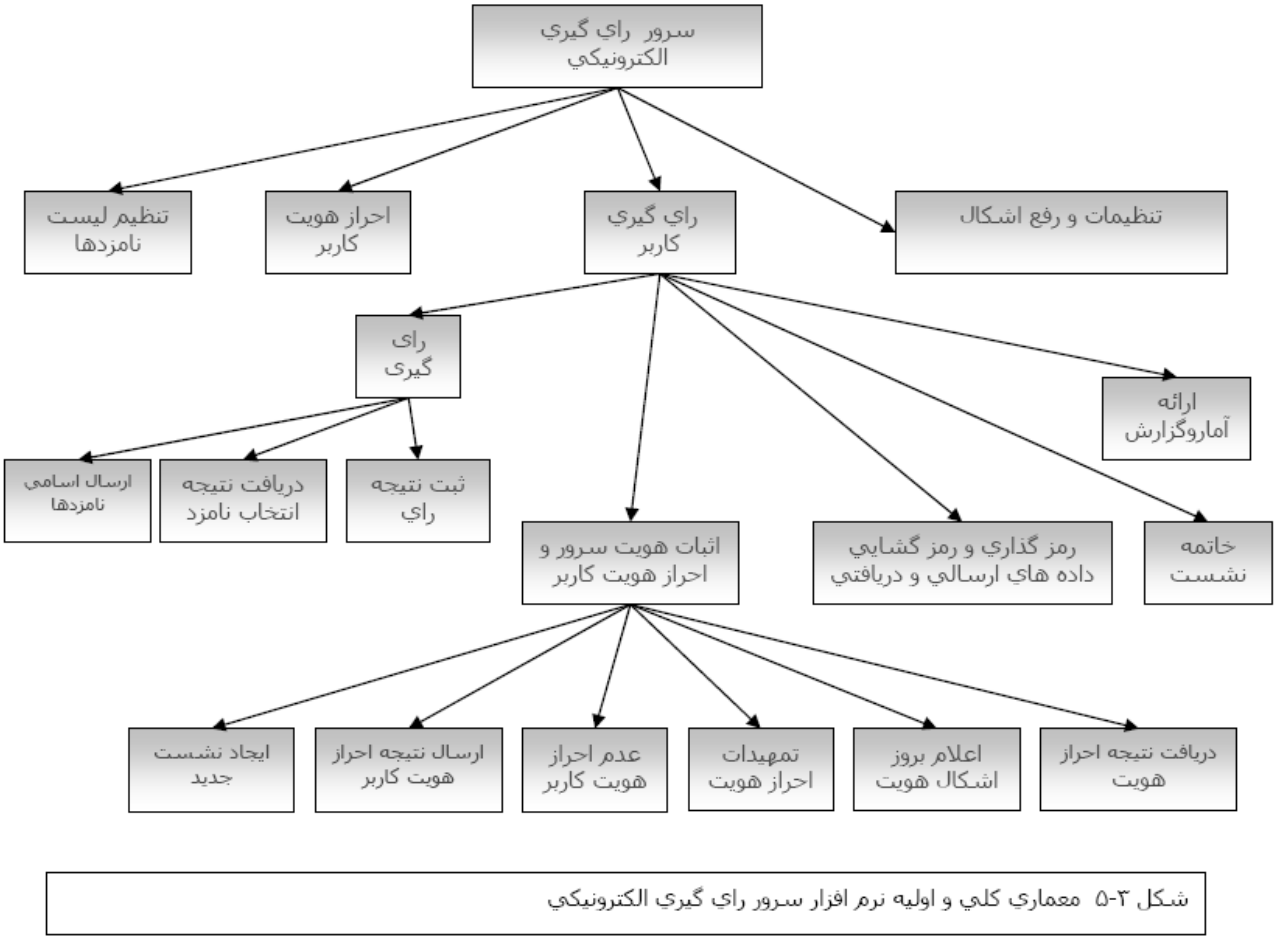
رای گیری

ارسال اسامی نامزدها

اسامی نامزدها که توسط Adminتنظـیم شـده و ذخیـره شـده اسـت از پایگـاه داده خوانده می شود و در قالب یک فایل XML به سمت کاربر ارسال می شود .

دریافت نتیجه انتخاب نامزد

نتیجه انجام رای گیری در قالب یک رای که ساختار آن در قسمت بعد تشریح می شود به سمت سرور فرستاده و سرور در این پیمانه آنرا دریافت می کند و رای را برای ثبت به پیمانه ثبت



شكل ۳-۵ معماري كلي و اوليه نرم افزار سرور راي گيري الكترونيكي

انتقال می دهد .

ثبت نتیجه رای گیری

در این پیمانه نتیجه رای گیری از کاربر در پایگاه داده آرا ثبت می شود .

ارائه آمار

پس از ثبت نتیجه را گیری کاربر ماری که بـا لحـاظ ایـن رای بدسـت آمـده از پایگاه داده دریافت و هم برای کاربر و هم برای صفحه پانل کنترل سرور ارسال می شود .

تنظیمات و رفع اشکال

در صورت بروز هر گونه اشکال admin شبکه مـی توانـد از طریـق پانـل کنتـرل تنظیمات را تغییر داده و اشکال را بر طرف کند . این پیمنه که به صورت همروند با پیمانه انتظار بـرای کاربر جدید و رای گیری کاربر اجرا می شود و تداخل عملیاتی با انها ندارد .

رمز گذاری و رمز گشایی داده های ارسالی و دریافتی

همانگونه که گفتیم پس از ایجاد session جدید ، با استفاده از کلید نشـست ایجـاد شده در مرحله احراز هویت کلیه داده های ارسالی و دریافتی در حین رای گیری از این پیمانـه کـه در واقع لایه آماده سازی (presentation) نرم افزار است عبـور مـی کنـد . تنظـیم الگوریتمهـای مـورد استفاده در این پیمانه در ابتدا توسط Admin سرور انجام می شود .

از آنجا که در حین طراحی تغییراتی در ساختار سلسله مراتبی پیمانه ها داده شد و شکل کلی معمـاری سرور بر اساس توضیحات تغیر نمود ، شکل نهایی معماری سلسله نراتب پیمانـه ای مناسـب سیـستم کارگزار (سرور) پس از ایجاد تغییرات به صورت زیر در می آید:

سرور رای گیری الکترونیکی

تنظیم لیست نامزدها و تنظیمات اولیه

انتظار برای کاربر جدید

رای گیری کاربر

احراز هویت کاربر و سرور

تمهیدات احراز هویت

ارسال نتیجه احراز هویت کاربر

دریافت نتیجه احراز هویت سرور

عدم احراز هویت کاربر

اعلام بروز اشکال هویت

ایجاد session جدید برای کاربر

رای گیری

ارسال اسامی نامزدها

دریافت نتیجه رای گیری

ثبت نتیجه رای

ارائه امار و گزارش

خاتمه session

رمز گذاری و رمز گشایی داده های ارسالی و دریافتی

بروز اشکال

تنظیمات و رفع اشکال

نرم افزار کاربر رای گیری الکترونیکی

براساس تغییراتی که در طراحی معماری سرور داده شد ، مناسب است که در طراحی نرم افزار مشتری نیز تغییرات لازم را لحاظ کنیم تا همگونی طراحی بین این دو به وجود آید . ساختار سلسله مراتب پیمانه ای تغییر یافته طراحی نرم افزار مشتری به شکل ۳-٦ در خواهد آمد .

مشتری رای گیری الکترونیکی

برقراری ارتباط

احراز هویت کاربر و سرور

تمهیدات احراز هویت

دریافت نتیجه احراز هویت کاربر

ارسال نتیجه احراز هویت سرور

عدم احراز هویت کاربر

اعلام بروز اشکال هویت سرور

ایجاد session

فرایند رای گیری

دریافت اسامی نامزدها

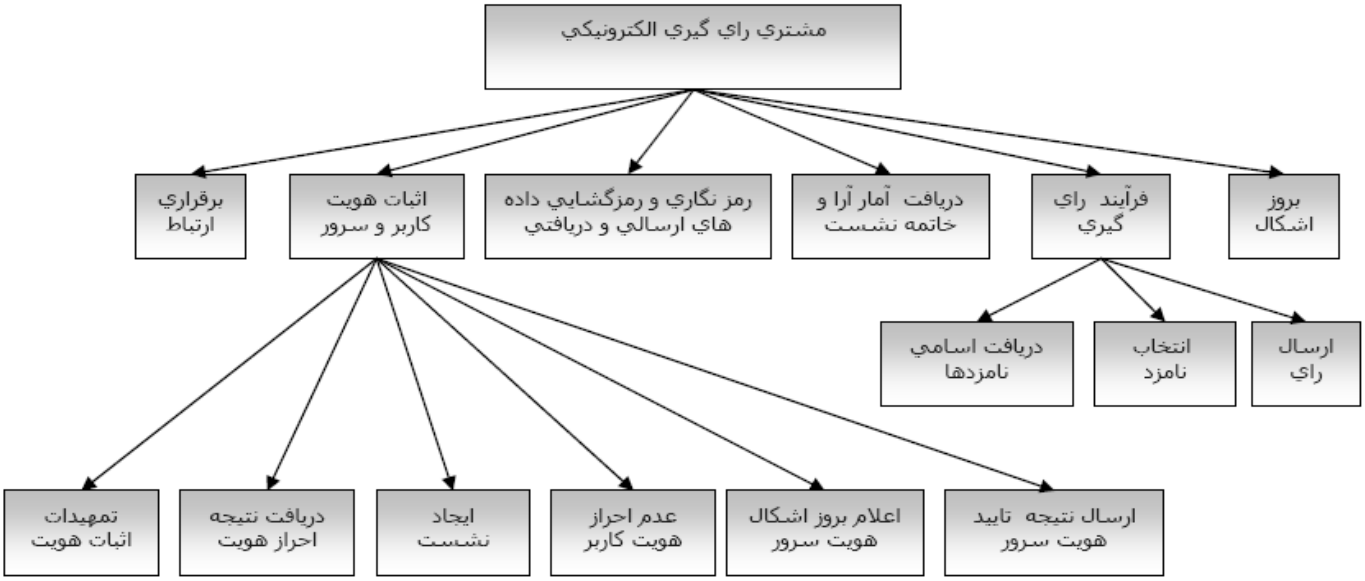
انتخاب نامزدها

ارسال نتیجه رای گیری (رای)

دریافت آمار آرا و خاتمه نشست

رمز نگاری و رمز گشایی داده های ارسالی و دریافتی

بروز اشکال



شکل ۳-۶ معماري كلي نرم افزارمشتري راي گيري الكترونيكي

مشتری رای گیری الکترونیکی

برقراری ارتباط و Negotiation اولیه

در این پیمانه با در خواست کاربر تقاضای برقراری ارتباط با سرور صورت می گیرد و در صورت پاسخ سرور یک ارتباط TCP بین مشتری و سرور برقرار می گردد . ضمن برقراری ارتباط مذاکره اولیه در مورد قواعد رمز گذاری و احراز هویت در بین مشتری و کارگزار رد و بدل می شود . (یا به عبارت بهتر توسط سرور ارسال و توسط مشتری تصدیق می شود .) این قسمت ساختمان داده خاص خود را می خواهد که هر دو طرف کدهای مخصوص قرارداد شده برای هر الگوریتم را بدانند .

احراز هویت کاربر و سرور

در پیمانه احراز هویت ابتدا پیمانه تمهیدات فراخوانی می شود و در تعامل با کاربر با استفاده از کلید خصوصی کاربر و سرور هر دو احراز هویت می شوند . الگوریتم احراز هویت در مرحله قبل ضمن مذاکرات اولیه تعیین شده است . پس از آن تابع دریافت نتیجه نتیجه احراز هویت کاربر را دریافت می کند . بسته به اینکه هویت تایید شده باشد یا نه مسیر جریان برنامه تغییر خواهد کرد . چنانچه هویت تایید شود ، کنترل به پیمانه ارسال نتیجه احراز هویت سرور و اگر تایید نشود ، به پیمانه عدم تایید هویت منتقل می شود . پیمانه عدم تایید هویت این نتیجه را به اطلاع کارر می رساند و کار را متوقف می کند .

اگرنتیجه احراز هویت سرور نیز مردود باشد کار به پیمانه بروز اشکال هویت سرور سپرده شده و ادامه کار متوقف می شود . تنها در صورتی که هر دو تایید هویت شوند ، مسیر اصلی برنامه به پیش می رود و نتایج دو پیمانه قبل به دست پیمانه ایجاد نشست می رسد تا با ترکیب انها کلید نشست را به وجود آورده نشست را آغاز کند .

جزئیات این امور در قسمتهای بعد مشخص خواهد شد . پیمانه هایی که در قسمت احراز هویت بکار گرفته می شوند : تمهیدات احراز هویت ، دریافت نتیجه احراز هویت کاربر ، ارسال نتیجه احراز هویت سرور ، عدم احراز هویت کاربر ، اعلام بروز اشکال هویت سرور و ایجاد session خواهند بود .

فرآیند رای گیری

در قسمت فرایند رای گیری ابتدا تقاضای ارسال اسامی نامزدها به سرور داده می شود و نتیجه دریافتی آن که فایل XML حاوی مشخصات نامزدهاست ، برای کاربر نمایش داده می شود . این فرایند با احضار پیمانه دریافت اسامی نامزدها صورت می پذیرد ، سپس پیمانه انتخاب نامزدها

فراخوانی می شود که طی آن نامزد از لیست اسامی که پیش روی او قرار دارد ، اسامی منتخبین خود را انتخاب می کند . این نتیجه در یک قالب مشخص به نام رای سازماندهی می شود و توسط پیمانه ارسال نتیجه رای گیری (رای) این رای بدست آمده به سمت سرور ارسال می شود .

دریافت آمار آرا و خاتمه نشست

پس از آنکه عملیات رای گیری خاتمه می یابد ، نتیجه انتخابات با لحاظ رایی که تازه ریخته شده برای کاربر نشان داده می شود . و با تایید کاربر کار خاتمه می پذیرد .

رمز نگاری و رمز گشایی داده های ارسالی و دریافتی

همانگونه که در گزارش سرور نیز توضیح داده شد ، کلیه نقل و انتقالات داده ای که پس از ایجاد نشست انجام می شود ف به صورت رمز شده خواهد بود ، این رمز نگاری و رمز گشایی در لایه ای جداگانه په در اینجا بهصورت یک پیمانه ارائه شده انجام می شود .

بروز اشکال

در صورت بروز اشکال در هر مرحله کنترل با ارسال کدی که نشان دهنده نوع اشکال است به این پیمانه منتقل شده و این پیمانه با نوع اشکال را به اطلاع کاربر می رساند .

تعریف واسط هر پیمانه

در این مرحله به تعریف واسطهای پیمانه ها می پردازیم :

```
interface evoting_server
{
    candidates ←     candidate_list_configuration ();
      algorithms ←      initial_configuration();
     new_user  ←      waiting_for_new_user();
                    user_voting( new_user  , algorithms , candidates);
                    problem_occur(prblm_code);
                    configure_managemnt();
}
```

```
interface  user_voting( new_user , algorithms , candidates  )
{
  session_key  ←  authentication( new_user , server , algorithms) ;
    result        ←  voting( new_user , session_key , candidates );
        (*  this result is not vote , it is a message that show voting succed  or
not. *)
          represent-statistics(session_key , new_user , server , //votes_DB );
          session_end(session_key);
  cryptotext   ←  cryptography(job, algorithms, plaintext );
}
interface    authentication( new_user , server , algorithms )
{
    result ←        authenticatian_alg( algorithms, new_user, server);
                    send_result( result) ;
    sar    ←        server_authentication_result_reseive();
                    user_authentication_problem( result);
                    server_authentication_problem( sar );
   session_key  ←  session_maker(result,sar);
}
 interface      voting( new_user , session_key , candidates )
 {

     res  ←  send_condidates_list( candidates);
      (*  res  show that module execute correct   *)
      vote  ←  reseive_voting_result(res);
                register_vote( vote ) ;
 }


 interface  evoting_client()
 {
      algorithms      ←  connect(server);
      session_key      ←  authentication( client , server, algorithms );
                          voting(server , client  , session_key);
                          statistic_receive( server . session_key);
                          end_session(server , session_key);
                          problem_occur( prblm_code);
    cryptotext         ←    cryptography(job , algorithms , plaintext);
 }
 interface authentication(client , server , algorithms)
```

```
{
    atn_res ← authentication_alg(algorithms, client , server);
    rac     ← receive_authentication_result();
                send_server_authentication_result(atn_res);
               client_auth_failed( rac );
               server_auth_failed( atn_res );
    session_key ← session_maker( rac , atn_res );
}
interface voting(servr , client , session_key)
{
candidate_list ← receive_candidate_information(server, session_key);
vote           ← voter(candidate_list);
                send_result( server , vote);
}
```

۳) توضیحات واسطها :

در واسط مولفه نخست که نشاندهنده صورت کلی سرویسهایی است که در نرم افـزار سـرور رای گیری ارائه می شود ، شکل کلی عملیاتی که باید صورت گیرد نشان داده شده است .

البته اعمالی که نشان داده شده به شکل ترتیبی پشت سر هم اجرا نمی شوند . این قالـب تنهـا صـورت کلی انها را نشان می دهد . شکل اجرای این فرایند ها در طراحی سطح مولفه مشخص می شود .

مولفه تنظیم لیست نامزد ها ، که نتیجه انجام آن به صورت یک لیست از نامزدها بازگردانده می شود ، نخستین مولفه این واسط است . تنظیمات اولیه که در تعامل با admin سرور صورت می گیرد و نوع الگوریتمهـای رمزنگـاری و احـراز هویـت در آن مـشخص مـی گـردد و در قالـب سـاختار داده ای algorithms بازگردانده می شود ، عمل دیگری است که انجام می گیرد . دیگـر مولفـه انتظـار بـرای کاربر جدید است که هربار پس از راه اندازی یک کاربر برای کاربر جدید انتظار را آغاز می کند .

اما اصلی ترین مولفه این قسمت مولفه رای گیریِ کاربر اسـت کـه از انچـه در مولفـه هـای قبلـی بـاز گردانده شده استفاده می کند . لیـست نامزدهـا ، مشخـصات الگوریتمهـا و مشخـصات کـاربر جدیـد ورودیهای آشکار این مرحله هستند .

```
interface evoting_server
{
    candidates ←    candidate_list_configuration ();
      algorithms ←     initial_configuration();
```

```
new_user ←      waiting_for_new_user();
                user_voting( new_user  , algorithms , candidates);
                problem_occur(prblm_code);
                configure_managemnt();
}
```

دو مولفه دیگر مولفه های بروز اشکال و مدیریت تنظیمات هستند که هریک در موقع مقتضی استفاده می شوند . ورودی مولفه بروز خطا کدی است ، که مشخصات کلی خطا را نشان می دهد و با استفاده از آن این مولفه خطا را handle می کند .

از بین مولفه های واسط اصلی تنها مولفه دیگری که مجموعه ای از سرویسهای کلی را ارائه می کند و براساس معماری نیاز به تشریح واسط دارد ، مولفه رای گیری کاربر است . واسط آن همانگونه که می بینید ، پنج سرویس کلی را ارائه می دهد .

```
interface  user_voting( new_user , algorithms , candidates  )
{
  session_key  ←  authentication( new_user , server , algorithms) ;
    result       ←  voting( new_user , session_key , candidates );
  (*  this result is not vote , it is a message that show voting succed  or not.
*)
                represent-statistics(  session_key,new_user  ,  server  ,
//votes_DB );
                session_end( session_key);
  cryptotext  ←  cryptography( job , algorithms, plaintext );
}
```

این سرویسها عبارتند از :

احراز هویت ، که با استفاده از الگوریتمها به احراز هویت کاربر و سرور رسیدگی می کند و نتیجه را در صورت توفیق به صورت کلید نشست به مولفه های بعدی تحویل می دهد .

رای گیری ،که با استفاده از مشخصات کاربر جدید ، کلید نشست و لیست کاندیداها رای گیری از کاربر را به انجام می رساند و رای را ثبت نموده و نتیجه را که نشان از توفیق یا عدم توفیق در انجام کار است به ما نشان می دهد که در تعیین قسمت بعدی کار ضروری است .

ارائه آمار که با دسترسی به پایگاه داده و استخراج نتایج از آن و ارسال این نتایج به کاربر و سرور برای update نمودن نتایج قبلی ارسال می شود .

مولفه ختم نشست که عملیات مربوط به ختم نشست را انجام می دهد . نکته این که چون در این مورد و مورد قبلی ارتباط با کاربر نیاز است به کلید نشست نیاز داریم .

رمزنگاری ، که متنی را به همراه الگوریتم رمز نگاری و نوع عملی که باید انجام شود (رمز نگاری یا رمز گشایی) دریافت می کند و نتیجه را که متن رمز شده یا از رمز در آمده است تحویل می دهد .

از بین این ها برای دو سرویس احراز هویت و رای گیری نیز واسط ارائه شده است .

واسط ارائه شده برای احراز هویت دقیقا به همان صورتی است که در ساختار پیمانه ای تشریح شده بود ، اما این ساختار هنوز نیاز به جرح و تعدیل زیادی دارد تا بتواند عملیاتی شود و احتمالا سرویسهای آن با هم تداخل و همروندی خواهند داشت تا بتواند احراز هویت امن را برای کاربر و سرور پدید آورد .

آنچه در اینجا می بینید شش سرویس است که عبارتند از تمهیدات احراز هویت در قالب مولفه (authentication_alg) که با استفاده از الگوریتمهای رمز نگاری احراز هویت طرف مقابل را انجام داده و نتیجه را در قالب result برمی گرداند . این نتیجه برای کاربر فرستاده می شود ، که در ضمن ان نیم کلید پیشنهادی سرور ارسال می شود و در مرحله بعد نتیجه احراز هویت کاربر نیم کلید کارر را برمی گرداند . دو مولفه بروز خطا برای پوشش خطاهای احتمالی در نظر گرفته شده است و در مولفه اخر با استفاده از دو نیم کلید ، کلید نشست تهیه می شود و به عنوان نتیجه باز گردانده می شود ، که همین نتیجه در کل نیز (برای کل مولفه احراز هویت) باز گردانده می شود .

```
interface    authentication( new_user , server , algorithms )
{
    result ←      authenticatian_alg( algorithms, new_user, server);
                  send_result( result) ;
    sar    ←      server_authentication_result_reseive();
                  user_authentication_problem( result);
                  server_authentication_problem( sar );
    session_key ← session_maker(result,sar);
}
```

در واسط انجام رای گیری نیز سه عمل کلی وجود دارد ، ارسال لیست کاندیداها به کاربر و نتیجه آن که نشان از موفقیت یا عدم موفقیت در کار مولفه دارد . دریافت نتیجه انجام رای گیری از نرم افزار مشتری و ثبت این رای در پایگه داده آراء .

```
interface     voting( new_user , session_key , candidates )
  {
```

```
        res  ← send_condidates_list( candidates);
        (* res  show that module execute correct   *)
        vote  ← reseive_voting_result(res);
                register_vote( vote ) ;
    }
```

اما در مورد واسطهای نرم افزار مشتری . یک واسط کلی که سرویسهای لازمبرای نرم افزار را مشخص می کند . که هفت سرویس کلی را نشان می دهد .

```
interface  evoting_client()
    {
        algorithms    ←  connect(server);
```

این سرویس ترتیب اتصال با سرور را می دهد و با انجام مذاکرات اولیه با سرور مشخصات الگوریتمهای مورد استفاده در احراز هویت و رمز نگاری را بر می گرداند .

```
        session_key    ←  authentication( client , server, algorithms );
```

کار کرد این مولفه همانند مولفه مشابه در نرم افزار سرور است و وظیفه تعامل با آن مولفه را در احراز هویت طرفین بر عهده دارد .

```
voting(server , client  , session_key);
```

این مولفه نیز کار اصلی رای گیری از کاربر را به انجام می رساند و در تعامل دو طرفه با کاربر و سرور کار رای گیری را انجام داده و نتیجه را به سرور ارسال می کند .

```
statistic_receive( server . session_key);
```

در مولفه دریافت آمار آراء در تعامل با سرور نتیجه رای گیری تا این لحظه از سرور دریافت می شود .

```
end_session(server , session_key);
```

کار های مربوط به ختم ارتباط در اینجا انجام می شود .

```
problem_occur( prblm_code);
```

این تابع نیز مشابه تابع همنام در سرور است ولی تنظیمات متفاوتی خواهند داشت و تفسیر کدها برای نرم افزار مشتری متفاوت خواهد بود .

```
        cryptotext    ←  cryptography(job , algorithms , plaintext);
```

این مولفه دقیقا با مولفه طرف سرور یکسان است و همان اعمال را در طرف کاربر انجام می دهد .

```
    }
```

توضیح واسط احراز هویت کاربر نیز مشابه احراز هویت سور است تنها تفاوت در ترتیب ارسال و دریافت نتایج احراز هویت است ، که در یکی در گام دوم و در دیگری در گام سوم انجام می شود .

```
    interface authentication(client , server , algorithms)
    {
```

```
atn_res ← authentication_alg(algorithms, client , server);
rac     ← receive_authentication_result();
          send_server_authentication_result(atn_res);
          client_auth_failed( rac );
          server_auth_failed( atn_res );
session_key ← session_maker( rac , atn_res );
}
```

کاری که در مولفه رای گیری انجام می شود بیش از این نیست که لیست کاندیدا ها از سرور دریافت شده و از طریق واسط کاربر ، به او نشان داده می شود و در تعامل با او انتخاب صورت می گیرد و نتیجه که در قالب رای (vote) برگردانده شده به طرف سرور ارسال می شود .

```
interface voting(servr , client , session_key)
{
candidate_list ← receive_candidate_information(server, session_key);
vote           ← voter(candidate_list);
                 send_result( server , vote);
}
```

٤) ساختمان داده های محلی و سراسری

لیست مشخصات کاندیدا ها (نامزد انتخاباتی)

ساختار فایل اسامی بصورت XML است. یک فایل نمونه از لیست مشخصات نامزدها در ادامه آمده است:

```
<?xml version="1.0"?>
<candidate-list>
    <candidate>
        <id>1</id>
        <name>
            <first>firstName1</first>
            <middle>middleName1</middle>
            <last>lastName1</last>
        </name>
        <title>First Candidate Title</title>
        <home-page>http://candidate1.homepage.chiz</home-
page>
```

```
            <picture>/pictures/candidate1.png</picture>
            <comment>This is CANDIDATE1 comment</comment>
        </candidate>
        <!-- LIST OF OTHER CANDIDATES HERE -->
</candidate-list>
```

تنظیمات (مشخصات الگوریتمهای مورد استفاده)

در این قسمت همانگونه که از اسمش پیداست مشخصات الگوریتمهای رمز نگاری و احراز هویت نشان داده می شود ، در این ساختار با استفاده از کدهای استاندارد برای الگوریتمها نوع آنها مشخص می شود .

مشخصات سرور

```
// This is a comment
server.port=2222
server.ip=127.0.0.1
db.ip=127.0.0.1
db.username=sa
db.passwd=secrete
db.catalog=evoting
db.driver=the.driver.class.Name
// End Of Setting!
```

ساختمان داده های برقراری ارتباط

رای دهنده

```
CREATE TABLE USERS(
	id integer  IDENTITY (1, 1) NOT NULL,
	firstname char(30) NOT NULL,
	lastname char(30) NOT NULL,
	publickey binary NOT NULL,
	PRIMARY KEY(id)
);
```

ساختمان داده های پیمانه احراز هویت

کلید عمومی و خصوصی

کلیدهای عمومی و خصوصی دو عدد باینری ۱۰۲٤ (یا ۲۰٤۸) بیتی هستند که توسط نـرم افزار های تولید کلید ، تولید می شوند . الگوریتم معمول برای استفاد از این رمز گذاری RSA است ، که در آن دو کلید مکمل هم هستند . متن رمز شده توسط یکی از این کلیدها ،فقط و تنها فقط بوسـیله کلید دیگر باز می شود . تنها کلیدهای عمومی در دسترس افراد هستند و کلیدهای خصوصی نزد خـود افراد نگهداری می شود . در اینجا ما از این رمز نگاری برای احراز هویت اشخاص استفاده مـی کنـیم . زیرا با ارسال پیام رمز شده با کلید عمومی تنها خود شخص قادر به گشودن ان پیام خواهد بـود . البتـه به دلیل کندی این رمز نگاری تنها در آغاز و برای احراز هویت وتعیین کلید نشست ازآن اسـتفاده مـی شود . و پس از تعیین کلید نشست (که مخفیانه با استفاده از کلید عمومی طرف مقابل ارسال می شود .) رمز گذاری به صورت متقارن انجام می شود .

نتیجه احراز هویت

```
<?xml version="1.0"?>
<authentication-result>
      <result>
            <message-id>1</message-id>
            <authenticator>
              <type> server </type>
              <ip> 127.0.0.1</ip>
              <port>2738</port>
              <public-key>110010…11</public-key>
            </authenticator>
            <authenticated>
              <type> client </type>
              <ip> 127.0.0.1</ip>
              <port>1092</port>
              <public-key>000111010111…1</public-key>
            </authenticated>

            <a_result> succed </a_result>
            <r-session-key>10011101001110…110</r-session-key>
            <crypto-final-session-key>not-recognized<crypto-     final-
session-key>
            <accept-session-key>not-recognized</accept-session-key>
            <comment></comment>
      </result>
</authentication-result>
```

کلید نشست

یک عدد باینری ۱۰۲٤ بیتی است که در فرایند احراز هویت با توافق طرفین تعیین می شـود و از آن پس در کلیه مرسولات بین دو طرف به عنوان کلید متقارن برای رمزنگاری استفاده می شود . این کلید روی خط به صورت رمز نگاری شده با استفاده از کلید عمومی طرف مقابل ارسال می شود .

رای

ساختار رای متشکل از

```
CREATE TABLE VOTES(
	userid integer NOT NULL,
	candidateid integer NOT NULL,
	PRIMARY KEY(userid, candidateid),
	FOREIGN KEY(userid) REFRENCES USERS.id
);
```

گزارش

```
<?xml version="1.0"?>
<candidate-list>
	<candidate>
		<id>1</id>
		<name>
			<first>firstName1</first>
			<middle>middleName1</middle>
			<last>lastName1</last>
		</name>
		<title>First Candidate Title</title>
		<home-page>http://candidate1.homepage.chiz</home-
page>
		<picture>/pictures/candidate1.png</picture>
		<comment>This is CANDIDATE1 comment</comment>
		<votes-number>1001</votes-number>
	</candidate>
	<!-- LIST OF OTHER CANDIDATES HERE -->
</candidate-list>
```

کد خطا

کد خطا شامل یک ساختار داده ای است که با استفاده از چهار عدد ۱۶ بیتی دو مشخصه را نشان می دهد عدد اول دسته بندی (category) خطا را مشخص می کند و عدد دوم نوع ان خطا را نشان می دهد . عدد سوم محل مقوع خطا را مشخص می سازد و عدد چهارم کدی است که مقصدی را که گزارش وقوع خطا به انجا باید ارسال شود را نشان می دهد . این اعداد توسط مولفه بروز خطا تفسیر و پیام خطا به محل مربوطه ارسال می شوند .

۵) محدودیت های طراحی معماری

٦) بازنگریها و اصلاحات

۵) تعریف مولفه ای پیمانه ها و طراحی سطح مولفه

ابتدا به تعریف مولفه های نرم افزار سرور رای گیری الکترونیکی می پردازیم .
با توجه به دید عملیاتی که از ابتدای طراحی آنرا دنبال نموده ایم ، در مولفه سرور عملیاتی را که باید انجام شود ، به صورت زیر خواهیم داشت . ابتدا تنظیم لیست نامزدها چنانکه در توضیحات این مولفه جداگانه خواهد امد انجام می شود . این کار در تعامل با admin سرور صورت می گیرد . سپس تنظیمات اولیه نیز توسط مولفه تنظیمات اولیه set می شود . در نهایت دو مولفه رای گیری و مدیریت تنظیمات همروند با هم آغاز می شوند .

```
 component evoting_server
 {
      candidates ←     candidate_list_configuration ();
        algorithms  ←    initial_configuration();
        finish        ←     true ;
 par_begin(  voter_handler( algorithms , candidates ) ,
configure_managemnt()  );
  }
```

در مولفه رای گیری کاربر دو عمل عمده وجود دارد :
انتظار برای کاربر جدید .
رای گیری کاربر .
finish یک آرگومان بولین است کف توسط مولفه مدیریت تنظیمات می تواند تغییر داده شود و true شدن آن به معنای اتمام ساعت رای گیری است . تا هنگامیکه این آرگومان false است ، کار handle نمودن رای دهندگان انجام می گیرد . این مولفه منتظر کاربر جدید می ماند و هنگامیکه کاربری ارتباط برقرار می کند ، کار رای گیری از او را در یک thread جداگانه به مولفه رای گیری کاربر می سپارد و خود در انتظار برای ورود کاربران جدید باقی می ماند . توضیح اینکه آرگومان

finish چون از طرف دو فرآیند همروند مورد استفاده قرار می گیرد باید دستیابی به آن در حریمهـای انحصار متقابل باشد .

```
component voter_handler( algorithms , candidates )
{
    while( finish ≠ true )
    {
        while( new_user ← waiting_for_new_user() = null ) ;
            (* do-nothing *)
        user_voting( new_user , algorithms , candidates);
    }
}
```

ماهیت ارتباط TCP که دارای توابعی برای انتظار برای درخواست ارتباط می باشد کمک می کند کـه این قسمت را براحتی با استفاده از یک ارتباط TCP پیاده سازی کنیم.

```
component waiting_for_new_user() : new_user
{
 (* with tcp connection functions we can easy implement this component
*)
}
```

رای گیری از کاربر در یک Thread جداگانه بوسیله مولفه ای که توصیف آن در زیر آمده انجام می شود . ابتدا بوسیله مولفه احراز هویت احراز هویت انجام می گیرد . سـپس رای گیـری ، ارائـه آمـار و ختم نشست انجام می شود . چنانچه در هریک از این مراحل اشکالی پیش آید ، مسئله به مولفه رخداد خطا ارجاع می شود .

```
component user_voting( new_user , algorithms , candidates )
 {
   try{

        session_key ← authentication( new_user , server , algorithms) ;

        result ← voting( new_user , session_key , candidates );
                represent-statistics( new_user , server , //votes_DB );
                session_end();
   }
   catch{
        problem_occur(code);
   }
```

```
}
```

در مولفه احراز هویت ...

```
component   authentication( new_user , server , algorithms )
{
    result ←      authenticatian_alg( algorithms, new_user, server);
                  send_result( result) ;
    sar    ←      server_authentication_result_reseive();
                  user_authentication_problem( result);
                  server_authentication_problem( sar );
   session_key ← session_maker(result,sar);
}
component  authenticatian_alg( algorithms, new_user, server): result
{
 }
 component  send_result( result) {}
 component  server_authentication_result_reseive() : sar{}
 component  user_authentication_problem( result){}
 component  server_authentication_problem( sar ){}
 component  session_maker(result,sar): session_key {}
```

مولفه رای گیری

درون مولفه رای گیری کارها به این صورت انجام میشود که ابتدا لیست کاندیداها بـه سـمت مـشتری فرستاده میشود . در صورت بروز اشکال ، تابع بروز اشکال اجرا خوهد شد .

سپس چنانچه نتیجه موفقیت آمیز باشد ، برای دریافت نتیجه صـبر گیـری منتظـر مـی شـود . پـس از دریافت نتیجه رای را مورد سنجش قرار می دهد و در صورتی که خرابی در آن بوجود نیامده باشد آنرا به ثبت می رساند . اما در صورتی که خرابی وجود داشته باشد دو راهکار در پیش روست ، یکی آنکه در داخل مولفه دریافت رای تا مادامیکه رای درست دریافت نشده تلاش چندین بـار صـورت گیـرد و دیگر اینکه کار به مولفه بروز اشکال سپرده شود تابا اطلاع دادن به کاربر ، ارتباط را قطع و کـاربر بـاز برای رای ریزی تلاش کند . هریک از راهکارها محدودیتها و نتایج خاصی دارد ، که باید با بررسی آنها راهکار بهتر انتخای شود .

```
component     voting( new_user , session_key , candidates )
{
   try{
          res  ← send_condidates_list( candidates);
```

```
        (*  res  show that module execute correct   *)
         if ( res = succed )
          {
              vote  ←  reseive_voting_result(res);
                   if( vote is validate )
                    {
                        register_vote( vote ) ;
                    }
                    else
                    {
                        code  =  vote_not_validate ;
                        problem_occue(code);
                    }
         }
         else
        {
                code  = send_condidate_list_not_succed_ ;
                 problem_occue(code);
         }
    }
   cach
   {
        problem_occue(code);
   }
}
```

مولفه رمز نگاری

در مولفه رمزنگاری ابتدا با بررسی آرگومان algorithms راه ادامه کار مشخص می شود و سپس کار رمز نگاری یا رمز گشایی صورت می گیرد . شکل کلی مولفه به صورت زیر خواهد بود . در هـر یـک از caseها یک الگوریتم که نوع آن مشخص شده برای کار مورد استفاده قرار می گیرد .

Job متعیر بولین ی است ، که نوع کار (رمزنگاری یا رمز گشایی) را مشخص می نماید .

```
component   cryptography(  job , algorithms ,  plaintext  ): cryptotext
 {
     if( job = encode)
     {
        switch to    algorithms   :
        {
         case  it is   RSA
```

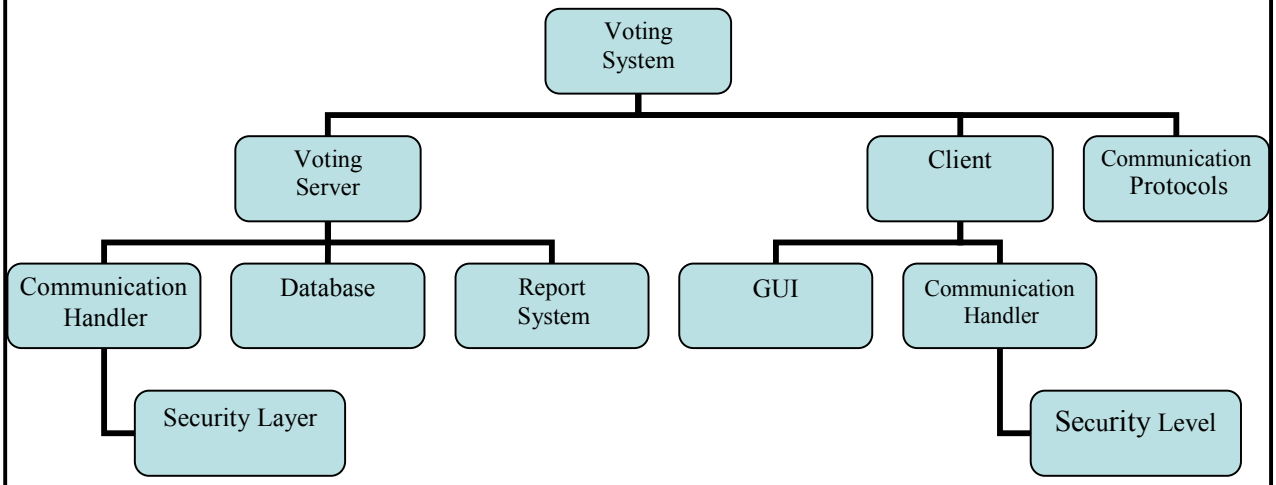
```
            begin
                cryptotext  = RSA( plaintext);
            end
        case  it is   cypher_text_mode_AES
            begin
                cryptotext  = cypher_text_mode_AES( plaintext);
            end
        ….
        case  it is   DES
            begin
                cryptotext  = DES( plaintext);
            end
        }
    }
    else   (* job = decode  *)
    {
      switch to    algorithms   :
      {
        case  it is   RSA
            begin
                cryptotext  = de_RSA( plaintext);
            end
        case  it is   cypher_text_mode_EAS
            begin
                cryptotext  = de_cypher_text_mode_AES ( plaintext);
            end
        ….
        case  it is   DES
            begin
                cryptotext  = de_DES( plaintext);
            end
      }
    }
}

component   send_condidates_list( candidates):res
{
}
component  reseive_voting_result(res):vote
{
```

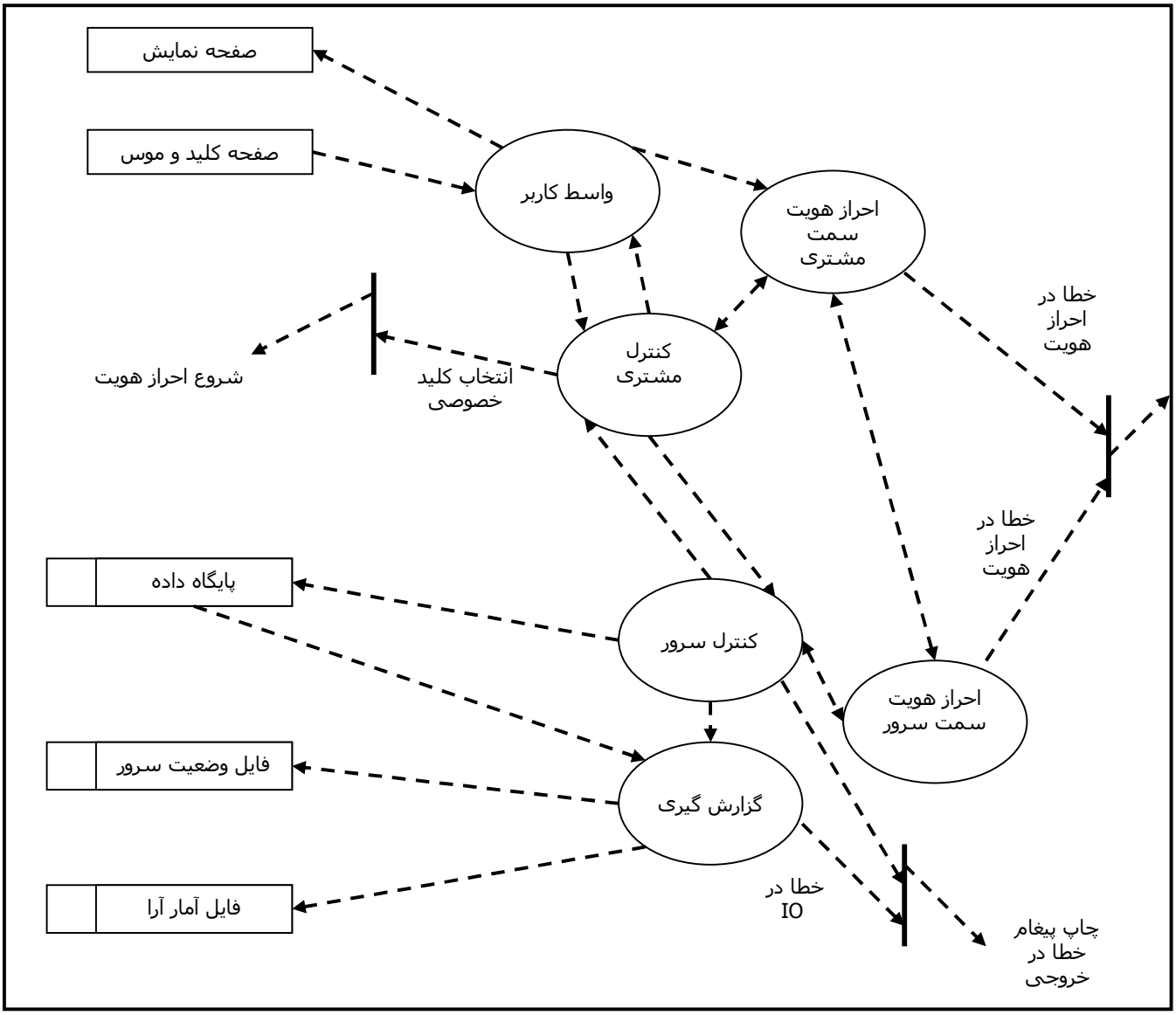
```
    }
  component  register_vote( vote )
   {
    }
component   represent-statistics( new_user , server , //votes_DB )
{
 }
component   session_end()
{
 }
component  configure_management()
{
}
component  problem_occur(prblm_code)
{
            switch to    prblm_code   :
            {
             case  it is  problem_27_0_6_2
                 begin
                        (*  dosomething *)
                 end
            ….
            case  it is   problem22_7_4_0
                 begin
                        (*  dosomething *)
                 end
            }
 }
```

فصل سوم

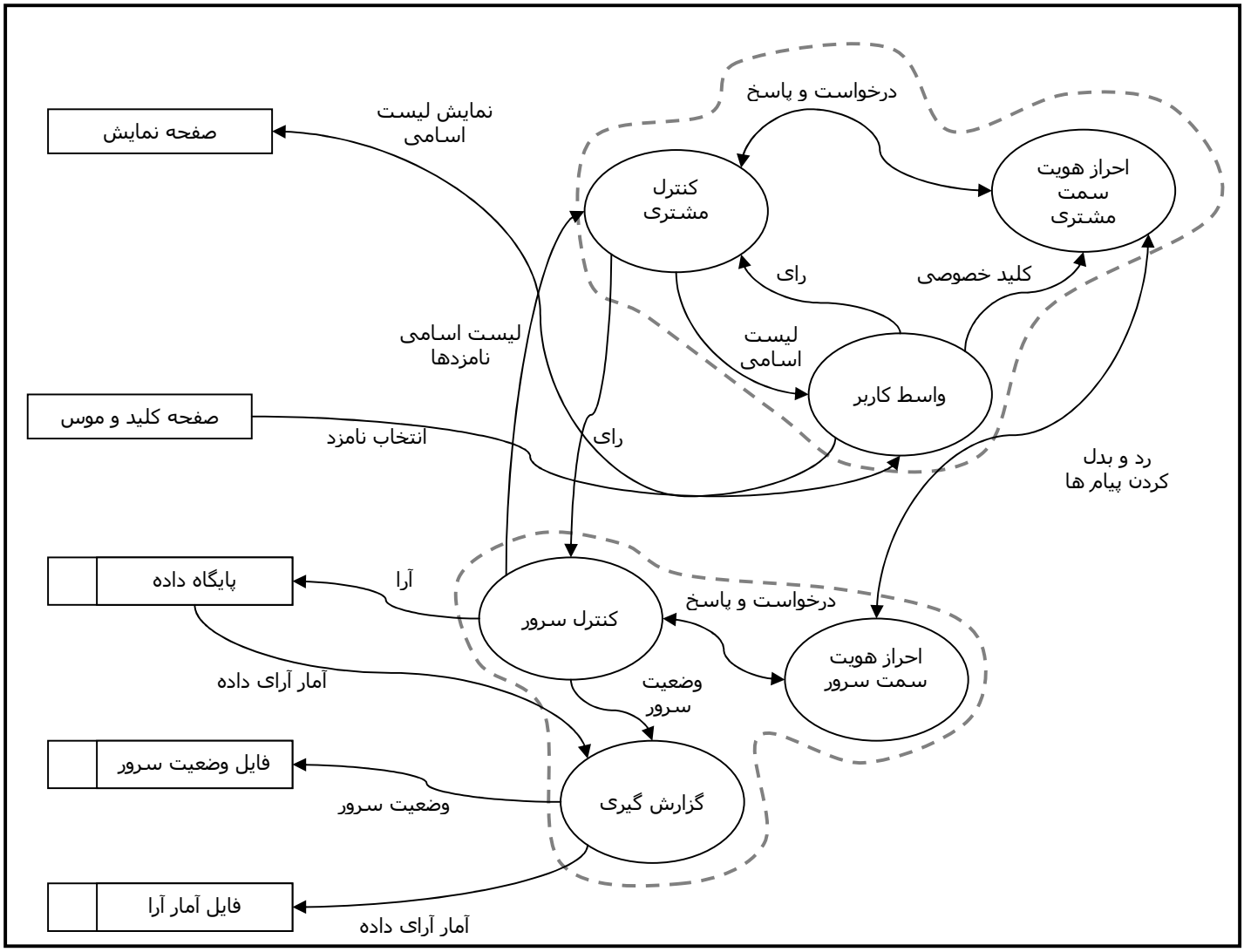
نمودارهای مختلف قسمت‌های تحلیل و طراحی و برخی مشخصه‌ها



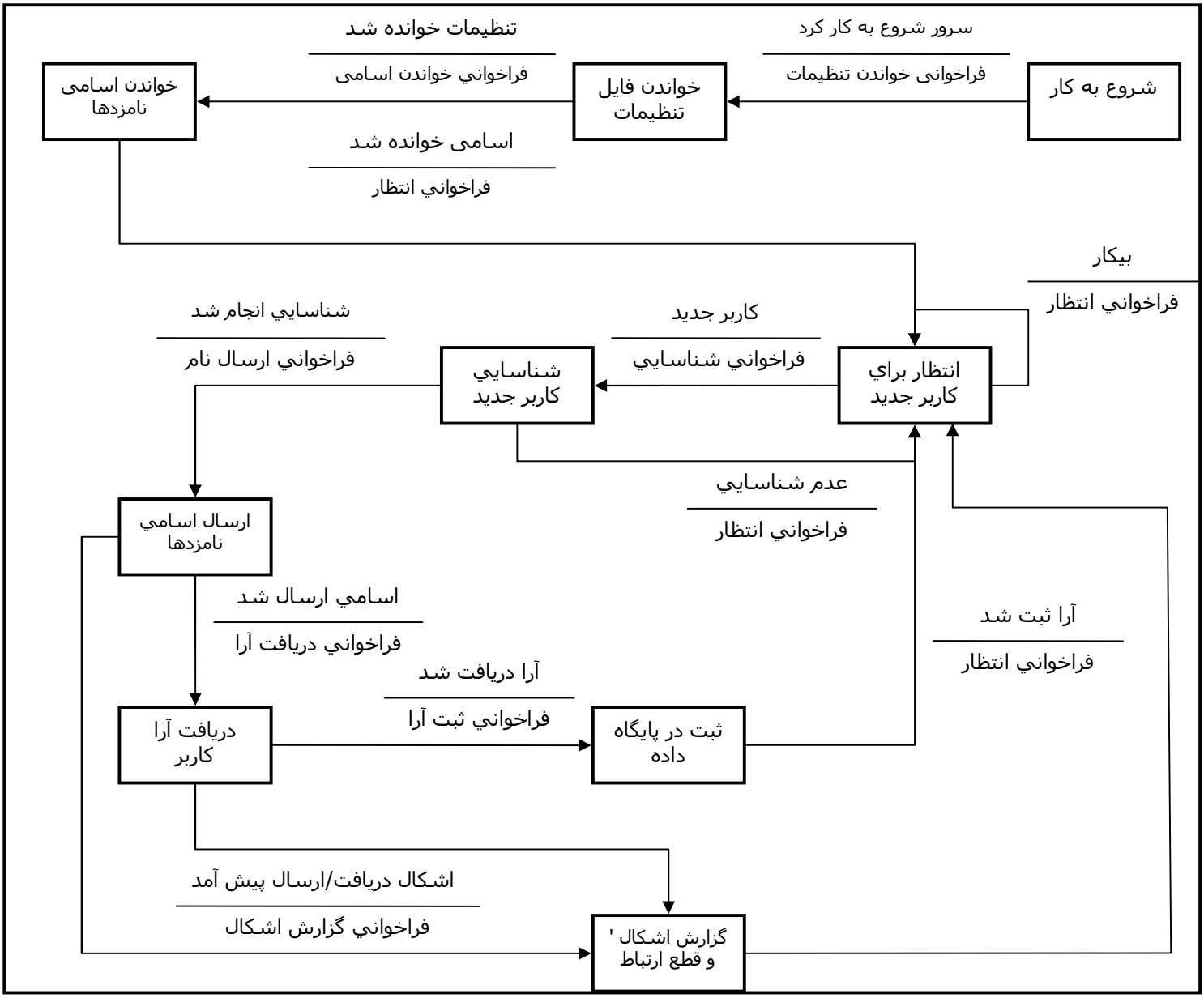
شکل ۰-اجزای مختلف سیستم رای گیری الکترونیک


صفحه نمایش
صفحه کلید و موس
واسط کاربر
احراز هویت
سمت
مشتری
خطا در
احراز
هویت
کنترل
مشتری
شروع احراز هویت
انتخاب کلید
خصوصی
خطا در
احراز
هویت
پایگاه داده
کنترل سرور
احراز هویت
سمت سرور
فایل وضعیت سرور
گزارش گیری
فایل آمار آرا
خطا در
IO
چاپ پیغام
خطا در
خروجی

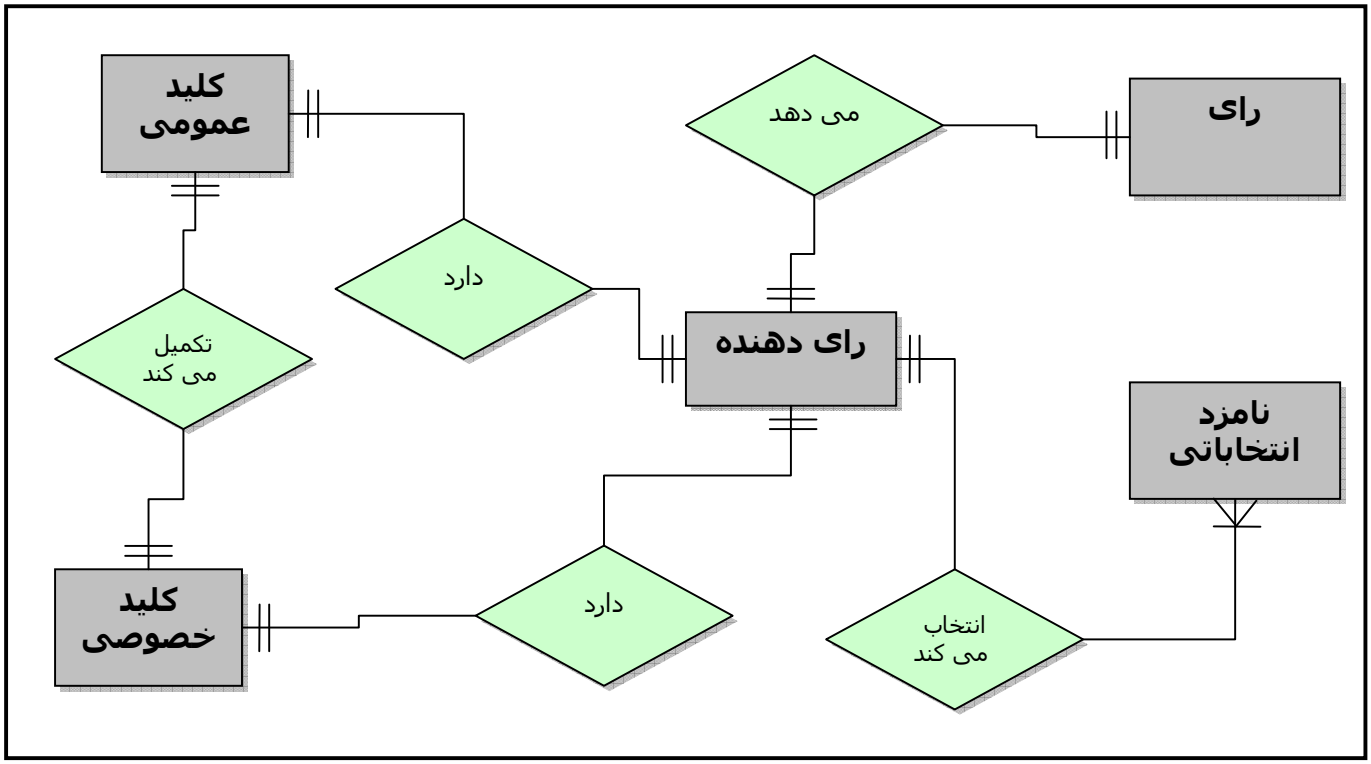
نمودار CFD


درخواست و پاسخ
صفحه نمایش
نمایش لیست
اسامی
کنترل
مشتری
احراز هویت
سمت
مشتری
رای
کلید خصوصی
لیست اسامی
نامزدها
لیست
اسامی
واسط کاربر
صفحه کلید و موس
انتخاب نامزد
رای
رد و بدل
کردن پیام ها
پایگاه داده
آرا
کنترل سرور
درخواست و پاسخ
احراز هویت
سمت سرور
آمار آرای داده
وضعیت
سرور
فایل وضعیت سرور
وضعیت سرور
گزارش گیری
فایل آمار آرا
آمار آرای داده

نمودار DFD


سرور شروع به کار کرد
فراخوانی خواندن تنظیمات
شروع به کار
خواندن فایل تنظیمات
تنظیمات خوانده شد
فراخواني خواندن اسامی
خواندن اسامی نامزدها
اسامی خوانده شد
فراخواني انتظار
بیکار
فراخواني انتظار
کاربر جدید
فراخواني شناسايي
انتظار براي کاربر جدید
شناسايي کاربر جدید
شناسايي انجام شد
فراخواني ارسال نام
عدم شناسايي
فراخواني انتظار
ارسال اسامي نامزدها
اسامي ارسال شد
فراخواني دریافت آرا
آرا ثبت شد
فراخواني انتظار
آرا دریافت شد
فراخواني ثبت آرا
دریافت آرا کاربر
ثبت در پایگاه داده
اشکال دریافت/ارسال پیش آمد
فراخواني گزارش اشکال
گزارش اشکال ' و قطع ارتباط

نمودار STD


کلید
عمومی
می دهد
رای
دارد
تکمیل
می کند
رای دهنده
نامزد
انتخاباتی
کلید
خصوصی
دارد
انتخاب
می کند

نمودار ERD

فصل چهارم

امنیت در
رای‌گیری الکترونیکی

"An observer of voting technology once remarked :`If you think technology can solve our voting problems, then you don't understand the problems and you don't understand the technology.' "

Dr. Rebecca Mercuri , Oct 2002 [1]

"اگر تصور می کنید که تکنولوژی می تواند مشکل رای گیری را حل کند ، یا مشکل رای گیری را درنیافته اید و یا تکنولوژی را نفهمیده اید ." (۱)

۱) مقدمه

در فصلهای گذشته به تحلیل و طراحی سیستم رای گیری الکترونیکی پرداختیم . از آنجا که امنیت در سیستم رای گیری الکترونیکی یکی از مهمترین مسائل در این سیستم قلمداد می شود . فصلی را جداگانه به این موضوع اختصاص داده ایم .

در بحث امنیت سیستم رای گیری الکترونیکی به دلیل آنکه با مباحث مختلفی روبرو می شویم ، موارد بسیاری که هر کدام بحثهای مربوط به خود را به دنبال دارند ، مطرح می شوند . مواردی که از سویی به علوم سیاسی و جامعه شناسی مربوط می شوند و بنیادهای فکری رای گیری را مطرح می کنند و از این نظر موارد امنیتی را تعیین می نمایند ، و از سویی دیگر مباحث مربوط به امنیت شبکه ها و امنیت در سیستمهای کامپیوتری را در بر دارند . همچنین مواردی از قبیل تراکنش های مربوط به انتقال داده ها و مشکلات هماهنگی سیستمها ، که مربوط به شاخه های دیگری از علوم کامپیوتر است نیز در اینجا به دلیل اهمیت سیستم و safety critical بودن آن (۲) اهمیت می یابند .

در کل مشکلات امنیتی این سیستم از چند موضع نشات می گیرند .

نخستین دلیل بروز مشکلات ، اشکالات ذاتی است که در ذات رای گیری نهفته است . مشکل آنجاست که به کار گیری سیستم اینترنتی نه تنها موجب رفع این موانع و اشکالات نمی شود ، بلکه به تشدید آنها منجر می گردد .

دومین دسته اشکالات اشکالاتی هستند که در ذات بستری که سیستم بر روی آن کار می کند ، وجود دارند ، این اشکالات هرگز از طریق برنامه قابل ترفیع نیستند و تنها راه حل برای غلبه بر آنها ایجاد بستری جدید است که فاقد مشکلات کنونی باشد . البته ناگفته نماند که برخی اشکالات موجود در بستر سیستم با بهبود طراحی و رفع اشکالات آن قابل ترفیع هستند .

دسته ای دیگر ناشی از مشکلاتی است که در مولفه های استفاده شده در پیاده سازی سیستم وجود داشته و یا در تلفیق آنها با یکدیگر بروز می کنند .

دسته ای دیگر از اشکالات اشکالاتی هستند که برای سیستم رای گیری الکترونیکی به شکل خاص بروز می کنند . بخشی از این اشکالات مربوط به طراحی سیستم و پیاده سازی آن می شود که با انجام

آزمایشات و تستهای مختلف و تلاش در جهت بهبود دادن طراحی سیستم قابل رفع هستند . و بخشی دیگر هم به دلیل شرایط خاص سیستم رای گیری الکترونیکی هرگز نمی توانند کاملا رفع شوند .

باید دانست ، امنیت سیستم رای گیری الکترونیکی و صحت و کارائی آن از اهمیت بالایی برخوردار است . بروز اشکالات جزئی در این سیستم می تواند به بحرانهای عمیقی تبدیل گردد و کانون توجهات قرار گیرد . این مسئله در توصیه نامه ای که از طرف گروه محققین آزمایش کننده نرم افزار Serve در امریکا تهیه شده به کرات مورد تاکید قرار گرفته است . (۳) مسئله ی رای گیری آنهم در ابعاد وسیعی چون رای گیری انتخابات مجلس و ریاست جمهوری مسئله ی بسیار مهمی است که وارد شدن خدشه به آن ، در اساس نظام دموکراتیک خدشه وارد می کند .
رای گیری یکی از مهمترین ضروریات یک سیستم سیاسی دموکراتیک است . بررسی تاریخچه دموکراسی در این زمینه می تواند در جهت شکل گیری ادامه بحث سودمند باشد .

۲) زمینه تاریخی و فلسفی رای گیری و دموکراسی (٤)

"Many forms of Government have been tried, and will be tried in this world of sin and woe. No one pretends that democracy is perfect or all-Wise. Indeed, it has been said that democracy is the worst form of Government except all those others that have been tried from time to time."
Winston Churchill, Nov 1947

در دموکراسی های امروز ، ما دموکراسی نمایندگان انتخابی داریم . بدان معنی که شهروندان با انتخاب نمایندگانی که تصمیم گیری برای آنها را بر عهده می گیرند ، بر خویش حکومت می کنند . دموکراسی از مبدا تاریخی- باستانی خویش در آتن یونان (۵) تا آنچه اینک بدان رسیده است ، از تغییرات شگرفی عبور کرده است . تعداد افرادی که واجد شرایط صلاحیت برای انتخاب در یک دموکراسی مدرن هستند بسیار بیشتر از افراد آتن تاریخی است . و به همین دلیل دموکراسی نمی تواند به شکل اصلی (حکومت مستقیم مردم بر مردم) آن در دول جدید ارائه و استفاده شود . نه تنها کشورهای مدرن بسیار وسیعتر از شهرهای یونان باستانی شده اند ، بلکه اقشار بسیار بیشتری از مردم اجازه رای یافته اند . نتیجتا ، جمع آوری آراء افراد در هر بحث برای یک موضوع دیگر ممکن نیست .

نمی توان در هر زمینه ای از همه افراد رای گیری به عمل آورد . و بـه همـین دلیـل مـا از دموکراسـی مستقیم (مردم بر مردم) به دموکراسی غیر مستقیم (نمایندگان) روی آورده ایم . در هر حال مفـاهیم اساسی که ، " دموکراسی را بهتر از همه انواع دیگر حکومت مورد آزمون قرار گرفته کرده اند ، سـاخته است . "، کماکان دست نخورده باقی مانده اند . واژه دموکراسی در اصل یک لغت یونـانی اسـت و از دو قسمت "demos" به معنی "مردم" و "kriatus" به معنی "اوتوریته" یا حکومت گری گرفته شده است .و مفهوما به معنی حکومت بوسیله مردم (government by the people) است . بنـابراین در دموکراسی این مردم هستند که تصمیم گیرنده اصلی برای حکومت انـد ، حـال اگـر نـه بـه شـکل مستقیم ، از طریق نمایندگان منتخب خود که آنها را برگزیده اند.

با توجه به آنچه در بالا آمد ، رای گیری در سیستمهای دموکراتیک از اصلی ترین ارکان تعیین کننـده برای حکومت مردم برمردم است و چنانچه در آن خدشه ای وارد شود ، خدشه کلی بـه ماهیـت نظـام دموکراتیک وارد آمده است . بنابراین همگان (حتی آنهایی که هیچ سر رشته ای از موضوعات امنیت و سیستمهای کامپیوتری ندارند .) حق دارند که نسبت به درسـتی کـارکرد چنـین سیـستمی از خـویش حساسیت بروز دهند و درستی کارکرد آن را زیر سوال ببرند .

۳) مشکلات ذاتی موجود در هر نوع رای گیری

عدم امکان تامین همزمان بازبینی رای و محرمانگی

بخشی از مشکلات رای گیری به دلیل ماهیت سیستم رای گیری و تناقضات موجـود در آن بـروز مـی نماید . برای مثال بدان دلیل که بحث در دست گیری قدرت در انتخابات مطرح می شود ، اولا همگـان انتظار دارند که رای آنها مخفی بماند و هرگز انتظار ندارند کسی بداند رایی که آنها به صندوق انداختـه اند چه بوده است . از طرفی به دلیل حساسیت موضوع مردم علاقه دارند که از این مورد اطمینان یابنـد که رای آنها هرگز مورد دستکاری قرار نگرفته و در نتایج نهایی موثر بوده است . این مورد دوم نیازمند آن است که هر فرد خود ممیزی رای خویش را بر عهده گیرد و از تاثیر آن در نتایج مطمـئن شـود . در ضمن به دلیل آنکه باید از رای ریزی چند باره افراد جلوگیری به عمل آید ، شناسایی آنها امری محـرز و غیر قابل حذف است . ناگفته پیداست که سه مورد اخیر هنگامی که در کنار هم بررسی شوند به یک تناقض منجر خواهند شد و هرگز برآوردن هر سه آنها همزمان میسر نخواهد شد . در این مورد سـعی

همه ی سیستمهای رای گیری بر آن است که تا آنجا که می توانند سیستم را بـه شـکلی در آورنـد کـه تمامی موارد فوق تا حدود زیادی تامین گردد و حداقل برای رای دهندگان راضی کننده باشد .(٦)

خرید و فروش رای

دومین مشکلی که در هر سیستم رای گیریی امکان بروز آن وجود دارد ، مـشکل خریـد و فـروش رای است . اما در سیستم رای گیری الکترونیکی به دلیل خواص و ماهیت آن این مشکل به شکل حـادتری مطرح می گردد .

"فروش آراء مسئله ای در همه رای گیریهاست ، اما در عرصه رای گیـری اینترنتـی بـه شـکل خـاص نگرانی جدی در این زمینه وجود دارد ، زیرا این خرید و فروش رای می تواند ، به شـکل اتوماتیـک ، در سطح گسترده ای صورت پذیرد ." (٧)

٤) مشکلات ناشی از بستر انتخاب شده برای سیستم

این مشکلات که شامل مشکلات موجود در بخشهای نرم افزارهای بـستر سیـستم ، سـخت افـزار ، شبکه ، ویروسها و ورم ها ، Spoofing و حمله فردی در میان راه و DoSمی باشـد ، در پیوسـت شماره ی دو به تفصیل توضیح داده شده اند . آنچه در آنجا آمده گرچـه صـرفا در مـورد نـرم افـزاری خاصی چون serve نگاشته شده ، اما مشکلات طرح شده عمومـا گریبـانگیر همـه ی نـرم افزارهـایی است که بخواهند در رای گیری الکترونیکی بکار گرفته شوند . در هر صـورت آنچـه در ایـن پیوسـت آمده صرفا خلاصه ای از مشکلات عمده ی این بخش است و بسیاری از جزئیات در آن نادیده گرفتـه شده اند . به جز حملات مذکور در این بخش ، روشهای دیگری نیز وجود دارد که پرداختن به آنها در این مقال نمی گنجد .

٥) مشکلات ناشی از اشکالات در طراحی

بخشی از مشکلاتی که در زمینه ی رای گیری الکترونیکی بروز می کند و امنیت آنرا با خطر مواجه می سازد ف به خاطر وجود مشکلات در طراحی سیستم رای گیری الکترونیکی است . چنانکه با رویکـرد امنیت گرایی و قائل شدن اهمیت فراوان برای این مسئله طراحـی صـورت نگرفتـه باشـد ، مـشکلات بسیاری برای ایمنی آن وجود خواهد داشت . اگر چه بخش فراوانی از مشکلات آنگونه کـه گفتـیم بـه دلیل طراحی نا امن بستری که این نرم افزار بر روی آن مستقر خواهد شد غیر قابل رفع هستند ، اما در

طراحی سیستم باید ، با لحاظ دامنه ی حملات ممکن علیه سیستم ، تا آنجا که ممکن است ، آنرا در برابر این حملات ایمن نماییم .

در این زمینه مشکلات پروژه رای گیری الکترونیکی حاضر از لحاظ امنیتی و راههای مورد نظر برای رفع آنها از طریق طراحی مناسب تر در بخشی جداگانه بررسی خواهد شد .

دلایل مخالفین توسعه سیستم رای گیری الکترونیکی

چنانچه پیوست ۲ این مقاله را بررسی کنید ف خلاصه ی مقاله ای است که متخصصین امنیتی برای بخشی از وزارت دفاع امریکا که مسئول توسعه پروژه رای گیری الکترونیکی است فرستاده اند و آنها را که مصمم به استفاده از نرم افزار Serve در انتخاباتهای آتی بوده اند ، از این کار بر حذر داشته اند . گویا درخواست آنان موثر افتاده و از ادامه ی رای گیری بوسیله ی این نرم افزار و نرم افزارهای مشابه در انتخاباتهای مهم جلو گیری به عمل آمده است .

دلایل این مخالفان کاملا مشخص است . این پروژه از لحاظ امنیتی چنان آسیب پذیر است که به کار گیری آن در زمینه ی حساسی مانند انتخابات می تواند آسیب های مهلک به بنیاد نهادهای دموکراتیک ، اعتماد عمومی و مشروعیت سیاسی منتخبین وارد سازد .

....

٦) دلایل موافقین توسعه سیستم رای گیری الکترونیکی

با وجود همهی موانع و مشکلاتی که در زمینه ی رای گیری الکترونیکی وجود دارد ، مزایای آن نیز قابل چشم پوشی نیست ، گرچه دلایلی که دولت ایلند در وصف سودمندیهای این پروژه ذکر می کند ، در برابر دلایل مخالفان طرح آن چنان نیست که رغبتی در استفاده از این طرح ایجاد کند و نتوانسته اهمیت وجود آنرا با دلایل مستحکم بیان دارد . (دلایلی که در آنجا ارائه می شود ، اینهاست : استفاده از آن آسانتر ، نتایج آن دقیقتر و فاقد آرای باطله است ، و در ضمن سرعت شمارش افزایش یافته و سیستم رای گیری مدرنیزه می شود . (۸))

اما با توجه به اینکه ایجاد یک سیستم ایمن رای گیری الکترونیکی قدم نخست در راه حرکت به سوی edemocrasy و دولت الکترونیکی است و وجود آن تامین کننده ی یکی از بنیادی ترین نیازها برای ایجاد نهادهای مدنی الکترونیکی است و از آنجا که حل مشکلات امنیتی این پروژه می تواند در پی ریزی ایمن تر بقیه ی اجزاء و نرم افزارهای لازم برای وصول به دولت الکترونیکی موثر باشد و راه پیش رو را نشان دهد و از آنجا که پرداختن به آن می تواند روندی را نشان دهد که شکل خاصی از

تراکنشها که نمی توانند به صورت عادی ، همانند سایر تراکنشها صورت پذیرند ، چگونه می تواند انجام شود ، از نظر من ادامه ی این پروژه و تلاش برای حل مشکلات آن نیازی است مهم و با وجود همه ی مخالفت ها نمی توان انرا واگذاشت و از نتایجی که در صورت موفقیت در پی خواهد داشت درگذشت.

۷) نیازهای کلی یک سیستم رای گیری الکترونیکی

....

در پیوست این مقاله یک طرح ، یک مقاله و منتخبی از مقاله ای دیگر آمده است . این سه به همراه منابع و مآخذی که در هر یک معرفی شده اند ، در مجموع راهنماهای مناسبی برای رسیدن به درکی صحیح تر در مورد امنیت در رای گیری الکترونیکی و اهمیت این موضوع برای کسانی که علاقه مند به پیگیری بیشتر آن هستند خواهد بود.

۸) نتیجه گیری

...

۹) منابع و ماخذ

[1] Dr. Rebecca Mercuri. A Better Ballot Box? *IEEE Spectrum Online*, October 2002.

[2] Electronic Voting: A Safety Critical System , Margaret McGaley, Dr. J. Paul Gibson , March 2003

[3] Jefferson, D.R., Rubin, A.D., Simons, B., and Wagner, D. *A Security Analysis of the Secure Electronic Registration and Voting Experiment (SERVE)*; www.servesecurityreport.org/.
[4] referece 2 , page 2

[5] Cynthia Farrar. *The Origins of Democratic Thinking - The invention of politics in classical Athens*.Cambridge University Press, 1988.

[6] California Secretary of State Ad Hoc Touchscreen Voting Task Force Report; www.ss.ca.gov/elections/taskforce_report.htm.

[7] reference 3 , page 2

[8] Making it Easier to Vote - Electronic Voting and Counting.
http://www.environ.ie/elecvote detail.pdf.

فصل پنجم

پیوست‌ها

۱) ترجمه طرح اولیه(proposal) پیشنهادی به دولت ایرلند برای رای گیری الکترونیکی که به بررسی نکاتی چند از موارد حساس امنیتی می پردازد .

رای گیری الکترونیکی : تحلیل موارد حساس ایمنی

موضوع تنظیم و انجام رای گیری الکترونیکی ، مستلزم ترکیبی یگانه از مهندسی تکنیکها و محاسبات (همویدگی ها) ، به همراه مسائل و مشکلات اجتماعیی است ، که محدودیتهای خاصی را بر سیستم ، در رای گیری مخفی و شمارش آراء ، تحمیل می کنند . با استفاده از کامپیوتر مساله آسیب پذیری کلی نسبت به محرمانگی (privacy) ، صحت (accuracy) ، و امنیت در فرآیند رای گیری ، مطرح می شود .

ایرلند و دیگر کشورهای اروپایی ، آغاز به تغییر تمام یا بخشی از سیستم رای گیری خود ، و اتوماسیون آن با استفاده از کامپیوتر نموده اند . به نظر می رسد که اروپا ، با وجود همه مشکلات تکنولوژی و اجتماعی جدی ، از جمله عدم امکان رسیدگی و نظارت ، افزایش تهدید رای فروشی ، تهدید استراق نتیجه و خطر حمله DoS ، که در مورد آن وجود دارد ، جهش خود را به سوی گسترش تکنولوژی رای گیری الکترونیکی (در عین دارا بودن این مشکلات) آغاز کرده است .

انواع مختلفی از سیستمهای رای گیری الکترونیکی مشغول به کار هستند ، کارهای قبلیی که در این زمینه صورت گرفته ، منجر به معرفی چند نوع سیستم رای گیری شده که هنوز مورد استفاده اند . هر یک از این سیستمها نیاز به وابینی های متعددی که برای تضمین صحت و جامعیت لازم است ، دارند

پیشنهاد ما این است که آزمودن انواع جدید سیستمهای رای گیری الکترونیکی برپایه پروژه تکامل تکنولوژیهای ارتباطی ادامه یابد و به سمت تولید مدل رسمی و ریاضی مسئله رای گیری پیش رود .

فرضیه اصلی این است که ، علاجی ریاضی وار ، مسئله رای گیری الکترونیکی را شایسته است ، که در حین ساخت مدلهای رسمی ، با استفاده از تکنولوژیهای روز علوم کامپیوتر و مهندسی نرم افزار ، قابل پی گیری است . هدف اصلی این تحقیق آن است که نشان دهد ، هیچ سیستم رای گیری نمی تواند ، تا زمانی که نیازهای اصلی این سیستم به گونه ای بهتر تعریف نشده اند و متدی برای پشتیبانی

از پیاده سازی و انتشار (deployment) سیستم رای گیری ایمن ، برپایه مدلهای نیاز رسمی تعیین نشده ، ادعا کند کاملا ایمن (safe) است .

استراتژی تحقیقات :

آزمودن تعدادی از سیستمهای رای گیری الکترونیکی موجود و انجام تحلیلهای ریسک ، در مورد عدم برآورده کردن نیازهایشان .

نشان دادن آنکه ، استفاده از روشهای رسمی (formal method) در مهندسی نرم افزار ، در چنین سیستمهایی ، می تواند ریسکهای معینی را کاهش دهد .

توسعه مجموعه ای عمومی و قابل گسترش (extensible) از متدهای رسمی و تکنیکها و ابزارهایی که می توانند در پیاده سازی سیستم جدید رای گیری ، بر پایه توسعه تکنولوژیهای ارتباطی ، به کار آیند .

دستیابی به درستی (validate) ، مدلهای تئوری ، در حین توسعه سیستم رای گیری از راه دور جدید ، که برپایه واکنشهای آن به مجموعه آزمایشات مربوط به نیازهای ایمنی ، درستی آن ثابت می گردد .

پروژه رهیافت (approach) استانداردِ بکارگیری متدهای رسمیِ مهندسی نرم افزار را دنبال می کند :

مبتنی بر یک سیکل تکراری تکاملی مدلهای تئوری با آزمون درستی آنها در برابر پیاده سازی واقعی و ملموس آن . این امر همانند پذیرش تکنیکهای مدلسازی شی گرا برای پروژه است ، به منظور آنکه تحلیلهای ترکیبی مدل ما در همه سطوح تجرید انجام گیرد .

این پروژه با پشتیبانی گروه تحقیقاتی TASS در NUI Maynooth ایرلند انجام می گیرد . این گروه هم اکنون همکاری خود را ، با DCU ایرلند ، Nancy فرانسه و Clemson امریکا ، در حوزه کاربردهای متدهای رسمی و سیستمهای حساس به ایمنی (safety critical) ، ادامه می دهند . گروه تحقیقاتی TASS سابقه همکاری صنعتی با شرکتهای مخابراتی (از جمله BT و France Telecom) را نیز در کارنامه خود دارد .

سفر به فرانسه و امریکا (برای همکاری بیشتر) می تواند در جهت تحقیقات ، در مراحل اولیه پروژه مفید باشد .

پیوست ۲

مقاله‌ی خلاصه‌ی ارزیابی گروه متخصصین کامپیوتر وامنیت از کار سیستم رای گیری SERVE (پروژه ۲۲ میلیون دلاری رای گیری الکترونیکی پنتاگون)

by David Jefferson, Aviel D. Rubin, Barbara Simons,*AND* David Wagner

ANALYZING INTERNET
VOTING SECURITY

ارزیابی‌ای گسترده پیرامون سیستم رای گیری اینترنتی

تجربه ثبت نام الکترونیکی امن و رای گیری اینترنتی (serve) ، سیستمی مبتنی بر اینترنت ، که بوسیله Accenture و پیمانکاران او برای FVAP (برنامه مشاوره رای گیری فدرال) وزارت دفاع امریکا ، انجام گرفت . ماموریت FVAP آن بود که موانع رای گیری الکترونیکی را ، برای همه شهرهایی که در طرح UOCAVA (All citizens covered by the Uniformed and Overseas Citizens Absentee Voting Act) قرار داشتند ، کاهش دهد . مشخصا شهروندان امریکایی که اعضای سرویس نظامی بودند ، و اعضای خانواده ی آنها و شهروندان امریکایی غیرحاضر در امریکا . serve قصد داشت امکان آنرا فراهم کند ، که این شهروندان هم توانایی ثبت نام در انتخابات و هم امکان رای دادن از طریق اینترنت را ، از هر کجای جهان ، داشته باشند . بدین معنی که ، این سیستم باید کامل ، بی نیاز از تست کردن و در هر حالت- تضمین شده باشد ، تا بتواند آراء واقعی را جمع آوری کند .

برای سهیم شدن ، رای دهنده ی واجد شرایط ، ابتدا در برنامه Serve نام نویسی می کند . پس از نام نویسی ، رای دهنده می تواند برای رای دادن ثبت نام نماید و در یک یا دو مرحله کوتاه ، از طریق هر کامپیوتری که یه اینترنت متصل باشد ، می تواند رای بدهد . البته سیستم عامل کامپیوتر باید ویندوز مایکروسافت و مرورگر وب هم باید IE یا netscape باشد . ضمنا مرورگر باید به گونه ای تنظیم گردد که قادر به دریافت اسکریپت های JavaScript ، چه از اسکریپت نویسی جاوا و چه از

اسکریپت نویسی اکتیو ایکس باشد و کوکی های نشست را هم بپذیرد . هیچ نرم افزار یا سخت افزار اضافی دیگری نیاز نیست .

هنگامی که یک شخص بصورت انلاین برای رای دادن ثبت نام می کند ، اطلاعات مربوط به او در سرور مرکزی وب ذخیره می شود ، تا بعدا بوسیله اداره محلی انتخابات (LEO) بازیابی گردد ، در هر نقطه LEO پایگاه داده اش را بروز می کند . زمانی که شخصی در انتخابات رای می دهد ، رای کاملا مخفی او در سرور مرکزی ذخیره می شود و در پایان توسط LEO ، که آراء را برای جمع آوری ، ذخیره کرده است ، دانلود می شود . ارتباط بین مرورگر وب کاربر و سرور مرکزی با استفاده از پروتکل SSL (لایه سوکت امن) محافظت می شود . هر بار که ارتباط جدیدی برپا می شود ، یک کنترل اکتیو ایکس بر روی PC رای دهنده دانلود می شود و برای تامین شکل عملیاتیی که در مرورگر فعلی در دسترس نبوده اجرا می گردد . [1]

علاوه بر آنکه استفاده کنندگان از سیستم آزمایشی serve ۲۰۰٤ تنها به پرسنل نظامی و شهروندان خارج از کشور مختص بود ، تنها استفاده از آن به هفت ایالت از پنجاه ایالت (آرکانزاس ، فلوریدا ، هاوایی ، کارولینای شمالی ، کارولینای جنوبی ، یوتا و واشینگتون) که سهیم شدن در آن را پذیرفته بودند ، محدود شد .

در انتخابات ۲۰۰٤ تعداد کسانی که جمع آوری رای آنها بوسیله serve انجام شد ، حدود ۱۰۰۰۰۰ رای دهنده (رقمی که با تعداد زای دهندگان یک ناحیه کوچک برابری می کند .) تخمین زده شد ، که در هر دو مرحله ابتدایی و انتخابات عمومی استفاده گردید . این رقم در مقایسه با حدود ۱۰۰ میلیون رای دهنده انتخابات عمومی سال ۲۰۰۰ ، یکی از اهداف این بود که تعیین شود ، آیا چنین سیستمی می تواند برای استفاده ی همه ٦ میلیون رای دهندگانِ UOCAVA مناسب باشد . با وجود همه محدودیتها ، serve یک سیستم واقعی است . سیستمهای مشابه serve ممکن است در آینده توسط Accenture و دیگر شرکتها ، برای استفاده عموم رای دهندگان به جای جمعیتی اینچنین محدود ، پیشنهاد شود . به همین دلیل بر مبنای تحلیلهایمان تصور می کنیم serve یمک آزمایش نیست ، بلکه سیستمی واقعی است ، که در سالهای آتی ، به شکل عمده ای ، می تواند بسط و توسعه یابد .

---

[1] نويسندگان چهار نفر از هشت متخصص كامپيوتر و امنيتي هستند كه برنامه ۲۲ ميليون دلاري پنتاگون (serve) ، پيرامون راي گيري اينترنتي ، را مورد بازبيني قرار دادند . در فاصله كوتاهي پس از انتشار گزارش كامل در ژانويه ۲۰۰۴ ، پنتاگون ، با در نظر گرفتن نگراني هاي امنيتي ، تصميم به عدم اجراي پروژه serve ، در انتخابات ۲۰۰۴ گرفت . هنوز امكان ارائه برنامه هايي همچون serve براي انتخابات هاي آتي وجود دارد . اين مقاله ، برگرفته از گزارش كامل ارائه شده مي باشد كه به تشريح پي آمد هاي امنيتيي كه نويسندگان در مورد serve تشخيص داده اند ، مي پردازد و بيشتر به مشكلات اجرا پذيري ِ راي گيري اينترنتي در شكل عام آن مي پردازد . براي سادگي ارائه ، زمان حال استفاده شده ، با وجود اينكه در حال حاضر ديگر طرحهاي زيادي كه از serve در انتخاباتي استفاده كنند ، وجود ندارد .

از آنجا که اینترنت مستقل از مرزبندی های ملیتی و سیاسی است ، انتخاباتی که اینترنتی برگزار گردد ، نسبت به حملات ، از هرکجای جهان ، آسیب پذیر است .

توصیه ما

در اینجا خلاصه ای از نتایج یافته های و توصیه های ما آورده شده است .

سیستم رای گیری الکترونیک مستقیم (dre) که ، به دلیل آسیب پذیریهای عمده دفاعی و امنیتی ، به شکل گسترده ای مورد انتقاد قرار گرفته ؛ که نرم افزارشان کاملا انحصاری و بسته است ، که به خاطر تصدیق و کیفیت از امنیت کافی برخوردار نیست ، که خود dre خصوصا نسبت به حملات داخلی (برنامه نویسی) متنوع و گسترده ای آسیب پذیر است ، و اینکه dre از هیچ امکانات و تمهیدات بازرسی و بازخوانی آراء (کاغذ یا چیز دیگر) که به شکل گسترده ای می تواند این مشکلات را شناسایی کند ، برخوردار نیست . همه این انتقادات که مستقیما متوجه dre بود ، می تواند در مورد serve نیز همانگونه مطرح شود . (٤)

به علاوه ، از آنجا که serve ، یک سیستم مبتنی بر اینترنت و تحت PC است ، شمار دیگری از مسائل بنیادی امنیتی نیز ، که گستره وسیعی از اشکال مختلف حملات شناخته شده (از جمله حملات DoS ، spoofing ، حملات ویروسی روی PC رای دهنده و ....) ، که هر یک از آنها می تواند فاجعه آمیز باشد ، را در بر می گیرد ، آنرا آسیب پذیر می نماید .

چنین حملاتی می تواند در سطح وسیعی روی دهد ، و توسط هر کسی در جهان می تواند انجام پذیرد ، دامنه وسیعی که از یک شخص ناراضی تنها گرفته ، تا یک نهاد دشمنی سرمایه دار که دور از دسترس قوانین امریکاست ، را دربر می گیرد . این حملات می تواند نتایجی در ابعاد وسیع داشته باشد ، صلب حقوق شهروندی رای دهنده ، نقض محرمانگی ، خرید و فروش رای ، جهت دهی آراء تا حد تغییر برآمد های انتخابات های مختلف ، که شامل انتخابات ریاست جمهوری نیز هست ، باشد . برخی از حملات می تواند در عین آنکه موفقیت آمیز است ، تشخیص داده نشده بماند . حتی اگر حملات تشخیص داده و خنثی گردد ، می تواند به صلب اعتماد عمومی نسبت به انتخابات بینجامد . تعیین احتمال موفقیت یک حمله ناممکن است ، اما ما نشان خواهیم داد که بیشتر حملاتی که ما در مورد آنها نگرانی داریم ، به آسانی می توانند صورت پذیرند .

در بعضی موارد ، ابزارهای در دسترسی در اینتر نت موجود است که می توانند به آسانی ، برای حمله به یک انتخابات ، تغییر داده یا استفاده شوند . و در ضمن توجه به این نکته ضروری است ، که

یکی از وسوسه انگیزترین اهداف در تاریخچه حملات اینترنتی ، انتخابات عمومی امریکا بوده است ، چه از روی اغراض کاملا سیاسی و چه به منظور خودبزرگسازی . آسیبهای ذکر شده ، بوسیله تغییر طراحی یا ترمیم باگهای Serve ، قابل تعمیر و ترمیم نیست .

این آ سیبها ، در بنیاد و معماری اینترنت و سخت افزار و نرم افزار PC ها ، که امروزه در همه عرصه ها حضور جدی دارند ، ریشه دارد . و به همین دلیل ، در آینده نزدیک (و بدون بازطراحی و جایگزینی قسمت عمده سیستم امنیتی سخت افزار و نرم افزار PC ها و اینترنت و برخی ریشه های شکافت امنیتی دیگر) ، قابل حذف و امحاء نخواهند بود .

ما شرایط متنوعی را بر روی serve آزمودیم ، در هر حال ، همه ی این آزمونها از انواع یکسانی از آسیب پذیری ها شکست خورد . متاسفانه ، ما نمی توانیم هیچ یک از آنها را معرفی کنیم . ما پیشنهاد می کنیم ، یک معماری کوشک - وار (kiosk) به عنوان نقطه آغاز طراحی سیستم رای گیری الکترونیکی با اهدافی مشابه SERVE ، اما بدون اطمینان و بنیان بر اینترنت و نرم افزارهای ناامن PC ، برای آن در نظر گرفته شود . (۳ را ببینید ، پیوست C)

یک دوره آزمایشی ظاهرا موفقیت آمیز در انتخابات ریاست جمهوری امریکا در ۷ ایالت ، در نظر بیشتر مردم ممکن است تجربه ای بزرگ و موفق برای اثبات قابلیت اعتماد ، اطمینان پذیری (robustness) و امنیت سیستم رای گیری serve باشد . چنین برآمدی می تواند ترغیب کننده ی توسعه نرم افزار بوسیله FVAP در انتخابات آینده باشد ، یا آنکه بازاری برای تولید محصولات مشابه توسط شرکتها برای انتخابات های تعیین قدرت در همه ایالات متحده و دیگر کشورها ایجاد کند .

در هر حال ، این واقعیت که هیچ حمله ی موفقیت آمیزی تشخیص داده نشده ، نشانگر عدم رخداد آن نیست . بسیاری از حملات ، خصوصا اگر هوشمندانه مخفی انجام شوند ، به شدت تشخیص شان سخت خواهد بود . حتی در مواردی که این حملات بر آمد انتخابات را تغییر می دهند نیز امکان وقوع چنین حملاتی هست .

A “successful” trial of SERVE in 2004 is the top of a slippery slope toward even more vulnerable systems in the future. (The existence of SERVE was cited as justification for Internet voting in the Michigan Democratic caucuses earlier this year.) Like the proponents of SERVE, we believe that there should be better support for voting for members of the military overseas. Still, because the danger of successful, large-scale attacks is so great, we reluctantly recommend shutting down the development of SERVE

immediately and not attempting anything like it in the future until the security problems of the PC and the Internet are resolved. The remainder of this article explains some of the reasoning behind these conclusions.

آسیب پذیریهای SERVE

به دلیل آنکه اینترنت ، مستقل از مرزبندیهای سیاسی و ملیتی است ، انتخاباتی که بر روی اینتذنت انجام شود از هر نقطه جهان آسیب پذیر خواهد بود . بنا بر این نه تنها احزاب سیاسی در تغییر نتیجه انتخابات با حمله به SERVE خواهند کوشید ، بلکه نفوذگران ، بزهکاران ، تروریست ها و حتی دیگر کشورها نیز در این رابطه تلاش خواهند کرد .

بدون در نظر گرفتن نقشه ها و سفسطه های مخالفان ، بسیاری از حملات توصیف شده می تواند ، توسط یک فرد ، با اتکا به تحصیلات پایه دانشگاهی در زمینه برنامه نویسی کامپیوتر ، انجام گیرد . در این قسمت به توصیفی کوتاه از حملات متنوعی که می تواند علیه SERVE صورت پذیرد ، آورده شده است .

عدم امکان بازبینی (ممیزی) رای بازخوانی شده و حملات داخلی

سیستمهای رای گیری غیر کاغذیِ DRE در ابعاد گسترده ای مورد نکوهش قرار گرفت ، زیرا امکان بازبینی آن وجود نداشت . هیچ راهی برای رای دهنده برای اطمینان از اینکه رایی که داده با رای او در داخل ماشین یکسان است ، و آنرا در صفحه نمایش ماشین ببینند . اگر متعاقبا مشکلات جدی در فرآیند جمع آوری آرا روی دهد (که همه ی آنها در تعداد زیادی اتفاق می افتد) ، هیچ راه مستقلی برای رد گیری ممیزی رای و کمک به حل مشکل وجود ندارد . بازبینی و تایید رای دهنده تنها راه آسان و موثر جلوگیری در برابر حملات داخلی برنامه است . همه ی گزینه هایی که در مورد نیاز به بازبینی رای و ردگیری ممیزی آن نیاز است ، و در DRE ایجاد شده بود ف بی تغییر در SERVE نیز به همان صورت ادامه یافته است . (www.notablesoftware.com/evote ، www.verifiedvoting.org ، و [1] را ببینید .)

محرمانگی

محرمانگی جمع آوری آرا در SERVE بوسیله استفاده از رمز نگاری تضمین شده است . هنگامی که رای اخذ می شود ، برای انتقال اینترنتی رمز گذاری می شود و در سرور مرکزی رمز گشایی می گردد .

هنگام دریافت ، رای از مشخصات رای دهنده جدا می شود و رای بی نشان باز رمز می گردد کـه تنهـا LEO ی ناحیه ی رای دهنده قادر به خواندن ان است . این آراء رمز شده ، در سرور مرکـزی ذخیـره می شوند و می توانند (در فرم ترتیب تصادفی) هرگاه توسط سرور در خواست شوند ، دانلود گردند . چنین معماریی چندین ریسک جدی برای محرمانگی به دنبال دارد . نخست آنکه ، LEO می توانـد دریابد که شهروندان ناحیه او به چه کسی رای داده اند . این عمل می تواند با دانلـود پـی در پـی آراء بخش از سرور مرکزی ، در هر بار انداخته شدن رای جدید ، و با توجه به نام دریافتی از رای دهنـده ، صورت گیرد . دوم اینکه ، شکل آراء ، که به صورت متن کاملا واضح دریافت می شد ، می تواند ایـن خطر را ایجاد کند که در سرور مرکزی ، نگهدارندگان (administrators) سیستم می تواننـد بداننـد که اشخاص چگونه رای داده اند . همچنین ، اگر ماشین serve بوسیله خرابکاران مـورد هجـوم قـرار گیرد ، محرمانگی کل آراء می تواند مخدوش شود . سوم آنکه ، نگهداری آراء رمز شده Serve بـرای هژده ماه یا بیشتر می تواند با خرابکاریها و یا افشای کلیدهای رمز سیستم مورد خدشه قرار گیرد .

خرید و فروش آراء

فروش آراء مسئله ای در همه رای گیریهاست ، اما در عرصه رای گیری اینترنتی به شکل خاص نگرانی جدی در این زمینه وجود دارد ، زیرا این خرید و فروش رای می تواند ، به شکل اتوماتیک ، در سـطح گسترده ای صورت پذیرد . در جریان انتخابات ریاست جمهوری ۲۰۰۰ ، چندین وب سایت ، برای در اختیار گذاشتن امکان تعویض رای گور/نادر برپایه ی یک سیستم نجیب (honor) که هـیچ پـولی در آن ردوبدل نمی شد ، ایجاد شدند . چنین روندی می تواند در مورد سیستم SERVE نیز بکار گرفته شود .و سرویسهای تعویض رای یا معامله ی آن ویا خرید آراء رای دهندگان SERVE ، مـی توانـد فراهم گردد .

روش مستقیم خرید رای ، شامل فروش اطلاعات شخصی و رمز عبور و کلید خصوصی شخصی است . یک راه دفاع ممکن می تواند این باشد که serve از رای دهی چند باره از طریق یک ادرس اینترنـت جلوگیری کند . اما این راه اصلا یک دفاع قوی نیست ، زیرا یک خریدار رای می تواند serv را چنـان فریب دهد که گویی از چند آدرس مختلف آراء را ارسال می کند و ثانیا به خاطر گـرفتن حـق رای از کاربران عادیی که از یک آدرس رای می دهند . راهی دیگر برای خرید رای از فروشـنده آن اسـت کـه نسخه ای از مولفه ی ActiveX ِ SERVE را مهیا کند که رای را طبق نظر خریـدار رای تغییـر مـی

دهد ، چنانکه نظر او تامین گردد . به نظر نمی رسد که SERVE راهی برای دفاع در مقابل چنین نوع حملاتی ، بتواند داشته باشد .

اثر شدید و عمیق ضربات – وقتی ری گیری ، به شکل کاغذی یا مکانیکی صورت مـی گیـرد ، تمـام تغییراتی که در رای گیری اتفاق می افتد در گستره ی کوچکی رخ می دهد ، اما اگر سیستم رای گیری الکترونیکی SERVE نسبت به انواع مختلفی از حملات آسیب پذیر باشد ، نقش درصد قابل توجهی از آراء اینترنتی ، آسیب پذیر خواهند بود . تنها یک گروه بد اندیش می تواند بـر ده هـا هـزار رای اثـر قابل توجهی اعمال کند . این در حالی است که هیچیک از تقلب های منفرد ، چنین اثر گـسترده ای را درسیستم های دستی ندارند .

جدولی که در زیر آمده ، خلاصـه ای از نتـایج برخـی آسـیب پـذیریهای مهمـی را کـه در SERVE تشخیص داده شده اند ، می باشد ، که در کنربرآورد ما درباره آن آسـیب پـذیری قـرار گرفتـه اسـت . جدول برای هر تهدید SERVE ، مهارتهای لازم برای حمله کننده ، مراحل یک حمله موفقیت آمیز ، اینکه حمله تا چه حدی واقع بینانه است ، و چه اقدام متقابلی در قبال آن می تواند صورت پـذیرد را نشان می دهد .

قسمت باقیمانده مقاله به توضیح درباره سـه آسـیب پـذیری مهـم در SERVE مـی پـردازد . بـرای توضیحات بیشتر می توانید به گزارش اصلی مراجعه کنید . (۲۰)

Threat	Skill needed	Consequences	Realistic?	Countermeasures
Denial of-service attack (various kinds)	low	disenfranchisement (possibly selective disenfranchisement)	Common on the Internet.	No simple tools; requires hours of work by network engineers; launchable from anywhere in the world.
Trojan horse attack on PC to prevent voting	low	disenfranchisement	There are a million ways to make a complex transaction such as voting fail.	Can mitigate risk with careful control of PC software; reason for failure may never be diagnosed.
On-screen electioneering	low	voter annoyance, frustration, distraction, improper influence	Trivial with today's Web.	Nothing voter can do to prevent it; requires new law.
Spoofing of SERVE (various kinds)	low	vote theft, privacy compromise, disenfranchised voters	Web spoofing is common and relatively easy.	None exists; likely to go undetected; launchable by anyone in the world.
Client tampering	low	disenfranchisement	One example: change permissions on cookie file. Many other trivial examples.	None exists for all possible mechanisms. Too difficult to anticipate all attacks; likely never diagnosed.
Insider attack on system servers	medium	complete compromise of election	Insider attacks are the most common, dangerous, and difficult to detect of all security violations.	None within SERVE architecture; voter-verified ballots needed; likely undetected.
Automated vote buying/selling	medium	disruption of democracy	Very realistic, since voter willingly participates.	None exists; buyers may be out of reach of U.S. law.
Coercion	medium	disruption of democracy	Harder to deploy than vote buying/selling, but man-in-the-middle attacks make it achievable with average skill.	None exists; likely to go undetected.
SERVE-specific virus	medium or high	vote theft, privacy compromise, disenfranchised voters	Some attacks require only experimentation with SERVE; others require leak of SERVE specs or code and resourceful attacker.	Virus-checking software can catch known viruses, but not new ones; likely to go undetected.
Trojan horse attack on PC to change votes or spy on them	high	vote theft, privacy compromise	Widely available spyware would be a good starting point.	Can mitigate risk with careful control of PC software; harder to control at cybercafe or other institutionally managed networks; likely to go undetected.

بزرگترین اسیب پذیریهای تشخیص داده شده در SERVE

عدم توان کنترل محیط (بستر) رای گیری

شاید بزرگترن چالش پیش روی رای گیری اینترنتی از اینجا ناشی می شود که مسئولین برگزاری انتخابات بر همه ی تجهیزاتی که توسط کاربران مورد استفاده قرار می گیرند ، کنترل ندارند .

از آنجا که رای دهندگان serve می توانند از طریق کامپیوترهای شخصی خود یا کامپیوترهای تحت کنترل دیگران رای دهند ، اشخاص(یا احزاب) ثالث می توانند کنترل تعداد وسیعی از کامپیوتر هایی را که برای رای دادن مورد استفاده قرار می گیرند ، را در دست گیرند .

چنین حملاتی می تواند در صورت فقدان مسائلی چون : محرمانگی رای دهنده ، عدم حق رای در انتخابات ، تغییر رای بدون هیچ کس (vote alteration without anyone) ، وجود مناصب رسمی انتخابات و رای دهندگان ، در مشاهده یا تشخیص هر مشکل ؛ بروز کند .

کامپیوترها . کامپیوترهای شخصی رای دهندگان ، بعید است آنگونه که می توانند با هم همکاری کنند ، بتوانند به دقت از خود دفاع کنند ، و به همین جهت سیستمهای رای دهندگان به شکل خاص در برابر حملات ، حساس (آسیب پذیر) هستند . حملات به آسانی می توانند ، تبدیل به حملات خودکار شوند ، نفوذگر ها می توانند به شکلی روال مند هزاران یا میلیونها کامپیوتر را جستجو کنند و آسانترین ها را برای نفوذ پیدا کنند . یک راه نسبتا آسان برای گرفتن حق رای دهی از رای دهندگان از کار انداختن ActiveX و یا کوکی های وب کامپیوتر آنهاست که در این صورت دیگر امکان رای دادن از طریق serve وجود نخواهد داشت .

به تناوب ، یک شخص ثالث خرابکار می تواند آراء غیر مجاز را طوری تغییر دهد که به نظر رسند از جانب یک رای دهنده ارسال شده اند . یک سیستم مشترک ، برای مثال کامپیوتر کتابخانه ای عمومی یا cybercafé ، عدم امنیت بیشتری دارد .

صاحب اصلی (owner) ، مسئول (administrator) ، یا حتی یک بیننده ی پیشین (prior visitor) می تواند یک نرم افزار جاسوس یا خرابکار (subversion) را قبلا روی کامپیوتر نصب کرده باشد . رای دهی در محل کار نیز می تواند چنین خطرات و ریسکهایی را در بر داشته باشد .

یک تحقیق نشان داد ، ٦٢ درصد از شرکتهای بزرگ امریکا بر ارتباط اینترنتی کامندان خود نظارت دارند و بیش از یک سوم آنها فایلهای روی کامپیوتر کارمندان خود را ذخیره و بازبینی می کنند . (اینجا را ببینید www.amanet.org/research/pdfs/ems_short2001.pdf)

نرم افزار . نرم افزارهایکاربردیی که قبلا نصب شده اند نیز ریسک بزرگی هستند . درهای پشتی در نرم افزارها وجود دارند و می توانند ، هنگام فعالیت کاربر برای دادن رای ، بدون اینکه دیده شوند ، بر فرایند رای گیری نظارت کنند و یا در آن خرابکاری اعمال کنند . آسیب پذیریهای امنیتی نرم افزار ، می تواند به نفوذگران راه دور اجازه دهد ، کنترل کاملی بر کامپیوتری که در نقطه ای دوردست قرار دارد ، داشته باشد . این کار می تواند با استفاده از نرم افزارهای کنترل از راه دور در دسترس ، همانند : PCAnywhere یا BackOrifice انجام گیرد . نفوذ موفق ، حتی در مورد کامپیوترهایی که از لحاظ دفاعی استوارند (well-defended) ، کاری روتین و عادی است .

ویروسها و کرمها . یکی از اشکال خطرناک حمله از دور ویروسها و کرمها هستند که خود را شیوع می دهند و با ظرفیتهای بد اندیشانه ای طراحی شده اند که می توانند ماشین راه دور را کنترل کنند و خرابی وسیعی را در انتخابات آینده بروز دهند .

از آنجا که ، نرم افزارهای بررسی ویروسی (virus-checking) تنها در برابـر ویروسـهای شـناخته شده ی قبلی دفاع می کنند ، بررسی کننده های ویروس اغلب در جلوگیری از بروز و شیوع ویروسـها و کرمهای جدید ناتواننـد . در سـال ۲۰۰۱ ، کـرم code red تنهـا در عـرض ١٤ سـاعت ، ٣٦٠٠٠٠ کامپیوتر را آلوده کرد . و در سـال ۲۰۰۳ کرمـی کـوچکتر بـسیاری از ماشـینهای ATM و میزبانهـای اینترنت را از کار انداخت . (اینجا را ببینید :

http://www.securityresponse.symantec.com/avcenter/venc/data/w32.sqlexp.worm.html) .

کرمهای جدید حتی بیش از این زهر آگین هستند ، اغلب با چندین روش خود را شیوع مـی دهنـد و قادرند از دیوارهای آتش (fire walls) و دیگر ابزارهای دفاعی عبور کنند و تحلیل شان هـم خیلـی مشکل است .

برای مثال ، زمان زیادی برای تشخیص اینکه SoBig.F یک اسب تروجان طراحی شده برای اسـتقرار موتورهای اسپم بود طول کشید . (اینجا را ببینید

http://www.securityresponse.symantec.com/avcenter/venc/data/w32.sobig.f@mm.html) . حمله کنندگان می توانند ویروسهای جدیدی بسازند ، یا به شکلی تغییرات لازم را بر ویروسهای فعلی اعمال کنند ، که از تشخیص داده شدن ، بر امان باشند . کیت (بسته لوازم) هـای ساخت ویروس در اینترنت کاملا در دسترس هستند . به علاوه ، حمله کنندگان از این نکته سـود مـی برند ، که می توانند نسخه ی جدید ویروس خـود را ، بـا اسـتفاده از نـرم افزارهـای عمـومی بررسـی ویروس در دسترس که عمده ی قربانیان حمله از انها استفاده می کنند ، آزمایش کنند و مطمئن شـوند که ویروس آنها پیش از انتشار تشخیص داده نخواهد شد .

وب سایتها . یک حمله ی پیوندی (hybrid) خطرناک ، قراردادن محتـوای بدخواهانـه در یـک وب سایت خاص است . برای مثال ، یک حمله کننده که با یک کاندیدا دشمنی دارد ، ممکن است در وب سایت ان کاندیدا تله ی پنهان (boody-trap) ی قرار دهد ، که بازدیـد کننـدگان سـایت آن کاندیـدا نتوانند با استفاده از serve رای بدهند . چنین جلوگیری از حق رای دادن هایی می توانـد صـدها یـا هزاران رای یک کاندیدا را کم کند ، که برای انتخاب مخالف او کافیست .

حمله های spoofing و man-in-the-middle

در حمله ی "فردی در بین راه" ، نفوذگر خود را در وسط ارتباط بین دو طرف قرار می دهد و بـرای هر طرف نقش طرف دیگر را بازی می کند . (خود را به جای طرف مقابل جا می زند .) برای سادگی بحث در حد این مقاله ف ما ابتدا بر راههایی که حمله ی فردی در بین راه می توانـد محرمـانگی رای

دهنده را به خطر اندازد ، توجه ویژه می کنیم ، اگر چه تکنیک های مشابه این حمله مـی توانـد بـرای دیگر انواع حملات ، (برای مثال خرید رای) استفاده شـود . اسـتفاده از SSL انـدکی مـی توانـد در تخفیف شدت این حملات بـه محرمـانگی مـوثر باشـد . هـر man-in-the-middle می تواند ، بی آنکه تغییری بروز دهد ، نقـش دروازه ی SSL ، بـرای رد کـردن داده هـای برنامه ی کاربردی بین رای دهنده و سرور رای گیری را بازی کند . حملـه کننـده مـی توانـد همـه ی ترافیک را با رمز گشایی و رمز گذاری دوباره ی داده ها ببیند ، و همانند ارتباطات عـادی بـین ایـن دو عمل کند . در شکل موثر ، می تواند با استاده از دو نشستِ SSL رابطه ی خود را برقرار کند ، یکی بین خود و رای دهنده و دیگری بین خـود و سـرور رای گیـری . و هـیچکس در نخواهـد یافـت کـه مشکلی این وسط وجود دارد . این نوع حملات محتمل اند ، زیرا مرورگر رای دهنده ، اطمینان حاصل نمی کند از اینکه با سرور واقعی در رابطه است یا نه . و تنها می داند که با کسی در ارتباط اسـت کـه دارای یک مجوز معتبر ssl است (که می تواند حمله کننده باشد .) حمله ی فردی در بـین راه حتـی می تواند در جلوگیری از رای دادن افراد ، با از کار انداختن کلیه ارتباطها با رای دهنده ، موفق باشـد . serve البته از راهکارهای حفاظتیی استفاده می کند ، اما آنها فرض می کنند که رای دهنده دقیقـا مـی داند از یک فرایند رای گیری چه انتظاری باید داشته باشد ؛ این بدان معنی است که حملـه کننـده مـی تواند یک رایند رای گیری ایجاد کند که رای گیرنده باور کند که آن واقعی است . به شکل مـشابهی ، رای دهندگان نیز می تاونند به شکلی هدایت شوند که تصور کنند ، ثبت رای آنها موفقیـت آمیـز بـوده است ، در حالیکه ، آنها در واقع به جای ثبت درست در سرور با خرابکار مستقیما در ارتباط هستند .

The voters would discover when attempting to vote that they were not registered, but at that point there might be nothing they could do to resolve the situation.

شاید جدی ترین پی آمد حمله ی فردی در بین راه این است که حمله کننده ها می توانند در انتخابات ، برای تقلب بوسیله ی spoofing سرور رای گیری و مشاهده ی رای هر فرد به خصوص ، بـه کـار گرفته شوند . اگر رای دلخواه حمله کننده باشد ، رای دهنده به سایت اصلی serve هدایت می شود و اگر رای دلخواه حمله کننده نباشد ، نشست رای گیری کاملا spoof می شود . در این مورد ، کـاربر تصور می کند رای خود را داده است ، اما در واقع رای او دریافت نشده و در شمارش نخواهد آمد .

WE EXPECT THAT DENIAL-OF-SERVICE ATTACKS COULD DISENFRANCHISE A SUBSTANTIAL FRACTION OF THE SERVE POPULATION, AND THERE SEEMS TO BE LITTLE THAT SERVE CAN DO TO DEFEND AGAINST SUCH ATTACKS.

حمله عدم پذیرش سرویس (DoS)

حملاتی که ، با فعالیتهای خرابکارانه ای ، چون سربار اضافه بر سرور وب انتخابات ، مانع از استفاده ی رای دهندگان واجد شرایط از سرور شود ، همگی با عنوان حمله ی DoS شناخته می شود . یک شکل زننده ی حمله ی DoS ، حمله ی عدم پذیرش سرویس توزیع شده (DDoS) است . در این سناریو ، یک حمله کننده نوعا کنترل تعداد زیادی کامپیوتر را ، با انتشار یک ویروس یا کرم ویژه ، در اختیار می گیرد . در اصطلاحات امنیت کامپیوتر ، این سیستمها غالبا به عنوان سیستمهای zombie یا پیرو (slave) خوانده می شوند . زیرا حمله کننده در پشت نرم افزاری که آنها را آلوده کرده پنهان می شود و باعث می شود این سیستمها کورکورانه فرامین او را اجرا کنند . ابزارهای خودکار برای اجرا نمودن حملات DdoS در اجتماع (community) نفوذگران حداقل از ۱۹۹۹ در روالخانه آنها قرار گرفته است . و نفوذگران به اسانی می توانند شبکه های بزرگی از کامپیوترهای zombie خود گرد آورند . در فوریه ی ۲۰۰۰ ، بزرگترین حملات DdoS علیه غالب وب سایتهای پربازدید ، از جمله CNN ,Yahoo , eBay ، پی ریزی و انجام شدند . بعدا کشف شد که این حملات پرزیان توسط یک نوجوان ، که از سال ۲۰۰۰ در خارج از امریکا ساکن بوده ، به تنهایی صورت گرفته است . از آن زمان حملات DDoS دیگر کار روتین شده است . یک مطالعه بیش از ده هزار حمله ی DoS را در طول سه هفته در سال ۲۰۰۱ ثبت کرده است . (٦)

کرم code red ، برای مثال ، دارای کدی است که حمله ی DDoS را بر روی سایت کاخ سفید صورت داد . (خوشبختانه این حمله در آخرین دقایق شکسته شد .) در سال ۲۰۰۳ ، یک انتخابات اینترنتی در کانادا با وقوع حمله ی DoS در روز انتخابات به هم ریخت . این مثالها مثالهای تک وتوک نیستند ، بسیار ساده می توان حملات DDoS را پی ریزی کرد . و مقصرین تنها در مورد استثنایی دستگیر می شوند . اگر یک حمله کننده حمله ی DoS ی را در ابعاد وسیع طرح ریزی کند ، به شکلی که بتواند سرور رای گیری Serve را در روز انتخابات غیر قابل دسترسی نماید ، می توان از اعتبار انتخابات برگزار شده با توجه به تاثیر عدم اخذ آراء تعداد زیاد رای دهندگان uocava زیر سوال برد . برهم ریختن متناوب امکان رای گیری در مناطقی خاص ، ممکن است نتیجه ی انتخابات را تغییر دهد . تشخیص بخشی از حملات انتخابی ممکن است ، امکان داشته باشد ، اما به راحتی نمی

توان به مسدود شدنهای جایگاههای رای گیری واکنش مناسب نشان داد . تصور ما بر اینست که حملات DoS می تواند بخش وسیعی از جمعیت SERVE را با تضییع حق رای مواجه کند ، و به نظر نمی رسد که SERVE توانایی جلوگیری از این چنین حملاتی را داشته باشد.

نتیجه

به دلیل محدودیت فضا ، در اینجا تنها به بخش اندکی از حملات ممکن پرداخته شد . این خملات بر پایه آسیب پذیریهای بنیادیِ معماری کامپیوترهای شخصی (برای مثال ، کد malicious) و یا اینترنت (حملاتی چون spoofing و DoS) هستند . و توسط هر کسی در جهان می توانند انجام گیرند و در بسیاری از موارد می توانند با موفقیت انجام شوند ، در حالیکه کاملا ناشناخته مانده اند . بنابراین ، آنچه ما نتیجه می گیریم آنست که به طور کلی رای گیری اینترنتی ، و در شکل خاص آن serve ، قادر نخواهند بود – در آینده نزدیک – در یک انتخابات واقعی ، امنیت لازم را برقرار کنند . (گزارش کامل (۳) را ببینید .)

References

1. *California Secretary of State Ad Hoc Touchscreen Voting Task Force Report*; www.ss.ca.gov/elections/taskforce_report.htm.

2. Houle, K.J. and Weaver, G.M. *Trends in Denial of Service Attack Technology*. Technical Report, CERT Coordination Center (Oct. 2001).

3. Jefferson, D.R., Rubin, A.D., Simons, B., and Wagner, D. *A Security Analysis of the Secure Electronic Registration and Voting Experiment (SERVE)*; www.servesecurityreport.org/.

4. Kohno, T., Stubblefield, A., Rubin, A.D., Wallach, D.S. Analysis of an electronic voting system. *IEEE Security and Privacy* (2004).

5. Moore, D., Paxson, V., Savage, S., Shannon, C., Staniford, S., and Weaver, N. Inside the Slammer worm. *IEEE Security and Privacy* (2003).

6. Moore, D., Voelker, G.M., and Savage, S. Inferring Internet denial-ofservice activity. *Usenix Security* (2001).

7. Staniford, S., Paxson, V., and Weaver, N. How to own the Internet in your spare time. *Usenix Security* (2002).

مشخصات نویسندگان :

David Jefferson (d_jefferson@yahoo.com) is a computer scientist at Lawrence Livermore National Laboratory chair of the Technical Advisory Board for the Secretary of State of California

Aviel D. Rubin (rubin@jhu.edu) is a professor of Computer Science and the technical director of the Information Security Institute at Johns Hopkins University.

Barbara Simons (simons@acm.org) is an independent technology consultant, retired from IBM Research, and a past president of ACM.

David Wagner (daw@cs.berkeley.edu) is an assistant professor of Computer Science at the University of California at Berkeley.

© 2004 ACM 0001-0782/04/1000

B.Sc. PROJECT

Supervised by
Dr. Mohammad Hossein Yektaii
Dr. Ehsan Malekian

Electronic Voting

Mohsen Mollanoori Shams, Mohsen Momeni

Department of Computer Engineering

Teacher Training University

Karaj, Iran

September 2005