

RAPPORT

Applicaton de Vote avec chiffrement OpenPGP

Auteur :
Boussihi OUSSAMA

March 13, 2025

Contents

Introduction	2
1 Cahier des charges	3
1.1 Pésentation du sujet et Analyse du contexte	3
1.1.1 Pésentation du sujet	3
1.1.2 Analyse du contexte	5
1.1.3 Analyse de l'existant	5
1.2 Analyse des besoins	6
1.2.1 Besoins fonctionnels	7
1.2.2 Besoins non fonctionnels	7
2 Rapport Technique	8
2.1 Étude des techniques de chiffrement et transmission sécurisée	8
2.1.1 Les méthodes de chiffrement	8
2.1.2 Utilisation de GnuPG	9
2.1.3 Transmission sécurisée des votes via emails	10
2.2 Conception	11
2.2.1 Architecture globale du système	11
2.2.2 Modélisation UML	12
2.2.3 Base de données et stockage sécurisé	14
2.3 Réalisation	15
2.3.1 Technologies et outils utilisés	15
2.3.2 Implémentation des fonctionnalités principales	16
2.3.3 Gestion de la sécurité	17
3 Résultats et manuel d'utilisation	20
3.1 Interface graphique d'un votant	20
3.2 Interface graphique du centre de comptage (CO)	21
3.3 Interface graphique du centre de dépouillement (DE) et Résultats	22
3.4 Envoi des emails	23
4 Conclusion	24
4.1 Évaluation des résultats obtenus	24
4.2 Perspectives d'amélioration	25
5 Bibliographie	26

Introduction

L'informatique a, de nos jours, envahi toutes les sphères de notre société : du cadre familial au monde professionnel, les outils numériques se sont imposés en un temps record et sont devenus indispensables à l'organisation de nos structures sociales. Des dossiers médicaux aux actes de naissance, en passant par les formulaires administratifs, notre vie quotidienne repose désormais sur ces technologies, qui offrent rapidité, accessibilité et impartialité.

Dans ce contexte, l'idée d'informatiser le système de scrutin s'est imposée comme une réponse évidente aux besoins grandissants de transparence, de sécurité et d'accessibilité. Dématérialiser le vote permettrait non seulement d'automatiser le processus électoral, mais aussi d'assurer une meilleure protection des données et une plus grande neutralité dans le dépouillement des résultats. Toutefois, si l'outil informatique présente des avantages certains, il soulève aussi des défis majeurs, notamment en matière de confidentialité, d'intégrité et d'authenticité des votes.

L'application que nous allons vous présenter vise à répondre à ces enjeux en proposant un système de vote électronique sécurisé et confidentiel. Pour garantir l'intégrité des votes et éviter toute falsification, nous avons mis en place un processus rigoureux basé sur le chiffrement asymétrique et la transmission sécurisée des bulletins. Un des éléments fondamentaux de notre solution repose sur l'utilisation de deux centres distincts : l'un chargé de vérifier l'identité des votants, et l'autre responsable du dépouillement des votes. Cette séparation permet d'assurer un anonymat total, empêchant ainsi toute tentative de corrélation entre l'identité du votant et son choix électoral.

De plus, pour renforcer la sécurité du processus, les votes sont envoyés sous forme de messages chiffrés via email, garantissant ainsi leur confidentialité et empêchant toute modification non autorisée. Ce mode de transmission assure également une traçabilité efficace et permet aux différents centres de traiter les votes de manière indépendante, sans risque d'altération des résultats.

Ce rapport vient clôturer notre projet en mettant en lumière les défis auxquels nous avons été confrontés, ainsi que les solutions mises en œuvre pour y répondre. Nous analyserons les besoins identifiés, les contraintes rencontrées et les différentes étapes de conception et de développement de notre application. Enfin, une évaluation critique du système sera présentée, mettant en avant ses forces, ses limites éventuelles et les perspectives d'amélioration possibles.

Nous espérons que vous prendrez autant de plaisir à découvrir notre projet que nous en avons eu à le concevoir et à le développer. En vous souhaitant une agréable lecture.

1 Cahier des charges

1.1 Pésentation du sujet et Analyse du contexte

1.1.1 Pésentation du sujet

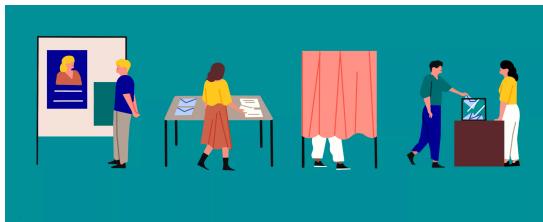


Figure 1: Système de vote traditionnel

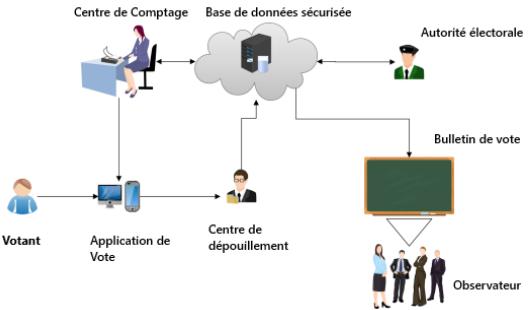


Figure 2: Système électronique de vote

Lors de toutes grandes élections, le votant doit se présenter lui même au lieu du scrutin et ainsi voter une unique fois. Chaque vote est alors anonyme et déposé dans une urne scellée. Le dépouillement s'effectue lorsque tout le monde a voté ou lorsqu'un organisateur le décide.

Le but du projet est de pouvoir modéliser ce système au travers d'une application et surtout de pouvoir tout automatiser. Chaque votant aurait alors un compte sur l'application et pourrait alors voter une unique fois. Le vote serait alors transmis au serveur qui ferait office d'urne en stockant les votes.

Le plus gros problème reste la sécurité car tous les votes sont anonymes. Il va donc falloir chiffrer les votes selon une méthode particulière telle que si un pirate intercepte un vote alors ce vote ne peut pas être lu. Contrairement au vote traditionnel, le vote ne sera pas anonyme car celui-ci sera chiffré. C'est pourquoi dans le but de montrer le fonctionnement de l'application, il sera intéressant que les utilisateurs puissent voir leurs votes chiffrés.

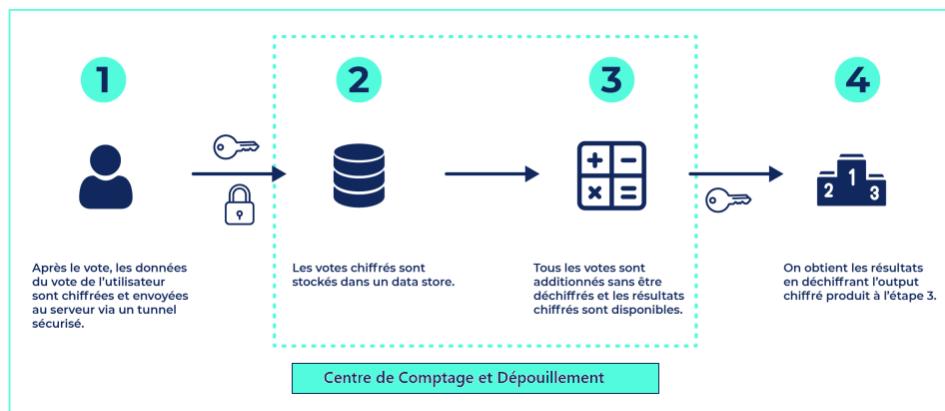


Figure 3: Modélisation d'un système avec chiffrement.

Il existe déjà de nombreux systèmes de vote électronique. Certains sont très sécurisés et d'autres le sont moins mais il est certain que c'est une technologie déjà présente sur le marché. Par exemple un employeur peut recourir au vote électronique pour les élections professionnelles dans le respect de certaines conditions et formalités préalables.

On trouve des avantages et des motivations pour adopter la démarche d'un système de vote électronique :

- **La rapidité du dépouillement de votes**, vu que les calculs sont faits plus rapidement par des ordinateurs, au contraire d'un dépouillement humain qui serait plus lent car il demanderait plus d'effort de la part des intervenants humains.
- **La fiabilité**, vu que l'application de vote électronique fait un calcul systématique des votes, au contraire d'un dépouillement humain qui pourrait avoir des défaillances.
- **La réduction de mobilisation de votants** : imaginons l'extension du vote électronique sur notre application pouvant s'exécuter depuis l'ordinateur du votant, ceci renforce sa participation vu qu'il pourrait voter depuis son ordinateur et n'aurait pas besoin de se déplacer.
- **Réduction des frais de matériel électoral et des besoins des ressources humaines**, sachant qu'il est parfois difficile de trouver des volontés pour devenir des assesseurs, faire voter et dépouiller.

Cependant, on se retrouve aussi avec des inconvénients liés à l'application de vote électronique :

- **Le code source de l'application de vote n'est pas partagé**, donc, les votants ne peuvent pas vérifier qu'il n'y a pas une faille à l'intérieur de l'application. La sécurité est gérée par le serveur de vote auquel les votants doivent faire confiance aveuglément.
- **Le risque de fraude est non négligeable**, il est impossible à l'électeur d'être sûr qu'il n'y a pas eu de fraude; par exemple, il ne pourrait pas savoir si le 8 serveur n'a pas autogénéré des votes ou s'il n'a pas effectué un vote à la place d'un votant.
- **Des risques de sécurité sont présents** : il faut employer des méthodes qui garantissent le chiffrement des votes pour qu'il ne soit pas connu, des méthodes pour vérifier le fait qu'un vote provient bien d'un votant qui n'a pas encore voté, donc garantir que les seules personnes qui peuvent voter, sont bien des personnes registrées qui n'ont pas encore voté. Il est aussi nécessaire d'employer des méthodes contre les attaques qui compromettent l'intégrité du système, par exemple, se protéger contre l'attaque de l'homme au milieu aussi appelée attaque de l'intercepteur*, cet attaque a pour but d'intercepter les communications entre deux parties, sans que ni l'une ni l'autre ne puisse se douter que le canal de communication entre elles a été compromis. Dans ce cas, l'attaque aurait lieu entre les votants et le serveur; de cette manière, l'authenticité des votes ne pourrait pas être garantie.

1.1.2 Analyse du contexte

Dans une Université, la direction d'un département souhaite organiser un vote à distance électronique par messagerie sécurisée afin d'élire un nouveau chef du département. Chaque membre votant doit pouvoir exprimer son choix de manière confidentielle, sans que son identité ne puisse être associée à son vote. Pour garantir cette confidentialité, le système mis en place repose sur un mécanisme de chiffrement asymétrique, où chaque vote est sécurisé avant son envoi. Ainsi, même si un tiers venait à intercepter les messages échangés, il serait dans l'incapacité de lire le contenu du bulletin de vote.

Contrairement aux systèmes classiques où un serveur unique centralise l'ensemble du processus électoral, notre approche repose sur deux entités distinctes : un **centre de comptage (CO)** et un **centre de dépouillement (DE)**. Cette séparation vise à renforcer l'anonymat du scrutin : CO est chargé de vérifier l'identité du votant et d'assurer qu'un même électeur ne puisse voter plusieurs fois, mais il ne peut en aucun cas accéder au choix du votant. De son côté, DE reçoit uniquement des votes chiffrés et les valide après déchiffrement, sans jamais connaître l'identité des électeurs. Ce fonctionnement empêche toute corrélation entre un électeur et son vote, garantissant ainsi une transparence et une impartialité totales.

De plus, pour assurer la transmission sécurisée des bulletins, les votes sont envoyés sous forme de messages chiffrés via email. Cela permet de créer une traçabilité tout en protégeant l'intégrité du processus électoral contre toute tentative de falsification. Le vote d'un électeur ne peut être ni modifié ni supprimé après son émission, et seul le centre de dépouillement, détenteur de la clé privée, peut accéder au contenu du scrutin.

Dans ce projet, nous avons donc cherché à concevoir une application de vote électronique qui apporte les avantages d'un système dématérialisé tout en garantissant une sécurité maximale. Nous nous concentrons particulièrement sur le chiffrement des votes et leur transmission sécurisée, tout en assurant une gestion rigoureuse des clés cryptographiques. Cependant, d'autres menaces, telles que l'usurpation d'identité du votant ou les failles liées aux terminaux d'accès, ne sont pas directement traitées dans notre solution.

1.1.3 Analyse de l'existant

Dans le cadre de notre projet, nous avons étudié plusieurs solutions de vote électronique existantes afin de mieux comprendre les approches déjà mises en place et d'identifier les éléments pouvant être intégrés ou améliorés dans notre propre application. Parmi ces solutions, nous avons découvert un logiciel particulièrement proche de notre concept : **Belenios**.

Belenios est une application de vote en ligne développée pour garantir un scrutin sécurisé et fiable, accessible via tout navigateur. Elle repose sur plusieurs principes fondamentaux visant à assurer la protection des votes et l'intégrité des résultats :

- **Sécurité** : Le vote est secret, aucun intermédiaire ne peut connaître l'identité et le choix d'un électeur.
- **Fiabilité** : Chaque votant peut vérifier si son vote a bien été pris en compte et chiffré. Un mécanisme de signature numérique est également utilisé pour garantir

que chaque vote provient bien d'un électeur authentifié.

L'utilisation de Belenios nécessite une authentification par un couple login/mot de passe spécifique à une élection, transmis aux électeurs via leurs adresses email. Le système repose sur un chiffrement asymétrique, notamment la méthode d'ElGamal, et exploite des protocoles avancés tels que la Preuve à divulgation nulle de connaissance (Zero Knowledge Proof) pour garantir la validité d'un vote sans divulguer d'informations sensibles.

Bien que Belenios soit une solution robuste et reconnue, elle n'est pas la seule sur le marché. D'autres applications de vote électronique existent avec des caractéristiques similaires ou complémentaires, notamment :

- **Helios** : Logiciel libre permettant un vote par "oui" ou "non".
- **Neovote** : Assure la sécurité des communications grâce au chiffrement via SSL/TLS.
- **Balotilo** : Met l'accent sur la simplicité en garantissant que les bulletins ne sont plus liés aux électeurs une fois l'élection terminée.

Notre application de vote sécurisé s'inspire de ces solutions existantes tout en y apportant des améliorations adaptées aux besoins spécifiques de notre projet. Contrairement à Belenios qui centralise la gestion du scrutin, nous avons adopté une architecture décentralisée en utilisant deux centres distincts (CO et DE). Cette approche permet d'assurer un anonymat total en empêchant qu'un seul serveur puisse associer l'identité du votant à son choix.

De plus, nous avons mis en place un système de transmission sécurisé des votes via emails chiffrés, garantissant que seules les entités autorisées puissent traiter et dépouiller les votes. Nous utilisons GnuPG pour le chiffrement asymétrique des bulletins, ce qui renforce encore davantage la confidentialité et l'intégrité du scrutin.

Le vote en ligne présente des avantages indéniables (rapidité, réduction des coûts d'organisation, accessibilité à distance), mais aussi des défis majeurs liés à la sécurité et aux risques de fraude. C'est pourquoi notre projet se concentre principalement sur l'application des protocoles de chiffrement avancés et des méthodes d'authentification sécurisée, afin de garantir un scrutin fiable, transparent et infalsifiable.

Ce projet prend ainsi tout son sens en nous permettant de nous initier aux protocoles d'échange sécurisé, tout en approfondissant nos connaissances sur les méthodes de chiffrement asymétrique et leur application dans un système de vote électronique fiable.

1.2 Analyse des besoins

L'analyse des besoins pour la mise en place d'un système de vote électronique sécurisé repose sur l'identification des attentes des différents acteurs du système. Ces acteurs sont le votant, le centre de comptage (CO), et le centre de dépouillement (DE). Les besoins fonctionnels se concentrent sur les exigences relatives aux actions que chaque acteur doit être capable de réaliser dans le cadre du processus électoral. Les besoins non fonctionnels, quant à eux, concernent les critères de performance, de sécurité, et de fiabilité du système dans son ensemble.

1.2.1 Besoins fonctionnels

Les besoins fonctionnels de ce système de vote électronique sécurisé se concentrent sur les actions essentielles que chaque acteur doit pouvoir réaliser. Pour le votant, il est primordial qu'il puisse soumettre son vote de manière sécurisée, sans que son identité ne soit associée à son choix. Il doit également recevoir une confirmation de la prise en compte de son vote. Le **centre de comptage (CO)**, quant à lui, doit être capable de vérifier l'identité de chaque votant afin de s'assurer qu'un même électeur ne vote qu'une seule fois, tout en garantissant l'anonymat de chaque bulletin. Une fois cette vérification effectuée, le **CO** doit transmettre le vote chiffré au **centre de dépouillement (DE)** pour qu'il puisse être déchiffré et validé. Le **DE**, enfin, doit pouvoir déchiffrer les votes de manière sécurisée et procéder à leur analyse pour déterminer les résultats du scrutin, sans jamais connaître l'identité des électeurs, assurant ainsi l'intégrité et la confidentialité du processus.

En tant que	Je veux	Afin de
Votant	Soumettre un vote de manière sécurisée Recevoir une confirmation de mon vote	Garantir la confidentialité et l'intégrité du vote. Être sûr que mon vote a été pris en compte correctement.
Centre de comptage (CO)	Vérifier l'identité des votants Transmettre le vote chiffré au centre de dépouillement (DE)	S'assurer qu'un même votant ne vote qu'une seule fois. Garantir l'anonymat des votants tout en assurant la traçabilité du vote.
Centre de dépouillement (DE)	Déchiffrer les votes Analyser et compter les votes	Valider la légitimité des votes reçus sans connaître l'identité des votants. Déterminer le résultat du scrutin de manière impartiale et transparente.

Table 1: Tableau des exigences fonctionnelles du système de vote

1.2.2 Besoins non fonctionnels

Les besoins non fonctionnels concernent les qualités du système, telles que la sécurité, la performance, la fiabilité et l'ergonomie. Ces critères sont cruciaux pour garantir que le système de vote électronique fonctionne de manière optimale, sans risques de manipulation ou de défaillance.

En tant que	Je veux	Afin de
Votant	Avoir une interface simple et intuitive pour voter	Faciliter l'accès au système de vote sans difficultés techniques.
Centre de comptage (CO)	Disposer d'un système fiable et sans erreur pour vérifier l'identité des votants Garantir la disponibilité du système en tout temps	Eviter toute fraude et garantir la transparence du processus électoral. Assurer une continuité du service sans interruption, même en période de forte affluence.
Centre de dépouillement (DE)	Avoir un mécanisme rapide et précis pour déchiffrer et valider les votes Assurer une traçabilité complète des votes	Optimiser le temps de traitement des résultats et garantir la précision des résultats. Garantir la vérifiabilité du système sans compromettre l'anonymat des votants.

Table 2: Tableau des exigences non fonctionnelles du système de vote

2 Rapport Technique

2.1 Étude des techniques de chiffrement et transmission sécurisée

2.1.1 Les méthodes de chiffrement

Dans cette section, nous présentons les principes fondamentaux des méthodes de chiffrement, tant symétriques qu'asymétriques, ainsi que le mécanisme de signature numérique assurant l'intégrité et l'authenticité des messages.

Chiffrement symétrique Le chiffrement symétrique repose sur l'utilisation d'une clé secrète partagée, notée K . Un message m est transformé en texte chiffré c par une fonction d'encryptage E :

$$c = E_K(m)$$

La déchiffrement se réalise à l'aide de la fonction inverse D avec la même clé K :

$$m = D_K(c)$$

La sécurité de cette méthode dépend de la confidentialité de la clé K et de la complexité du problème inverse, c'est-à-dire, la difficulté pour un attaquant de retrouver m à partir de c sans connaître K .

Chiffrement asymétrique Le chiffrement asymétrique utilise une paire de clés complémentaires : une clé publique K_{pub} et une clé privée K_{priv} . Un message m est chiffré avec la clé publique selon :

$$c = E_{K_{\text{pub}}}(m)$$

et ne peut être déchiffré qu'avec la clé privée correspondante :

$$m = D_{K_{\text{priv}}}(c)$$

Par exemple, dans le schéma RSA, la clé publique est constituée du couple (n, e) et la clé privée du couple (n, d) où $n = p \times q$ (avec p et q premiers) et

$$e \cdot d \equiv 1 \pmod{\phi(n)} \quad \text{avec} \quad \phi(n) = (p-1)(q-1).$$

L'encryptage se fait par :

$$c \equiv m^e \pmod{n},$$

et le déchiffrement par :

$$m \equiv c^d \pmod{n}.$$

Signature numérique La signature numérique garantit l'authenticité et l'intégrité d'un message. Pour signer un message m , l'expéditeur calcule d'abord un haché cryptographique $H(m)$ (par exemple, en utilisant SHA-256), puis chiffre ce haché avec sa clé privée :

$$s = \text{Sign}_{K_{\text{priv}}}(H(m)).$$

Le destinataire, en utilisant la clé publique de l'expéditeur, peut vérifier la signature en comparant le haché obtenu à partir de s avec celui de m .

2.1.2 Utilisation de GnuPG

GnuPG est une implémentation libre de la norme **OpenPGP**, qui intègre à la fois des mécanismes de chiffrement symétrique et asymétrique ainsi que la signature numérique. Dans notre projet, GnuPG est utilisé pour sécuriser les votes de manière robuste.

Approche hybride de chiffrement GnuPG adopte une approche hybride, combinant les avantages des deux types de chiffrement :

1. **Génération d'une clé de session** : Une clé de session aléatoire K_{session} est générée pour chaque opération de chiffrement.
2. **Chiffrement symétrique du message** : Le message m est chiffré avec cette clé de session :

$$c = E_{K_{\text{session}}}(m).$$

3. **Chiffrement asymétrique de la clé de session** : La clé de session est ensuite chiffrée avec la clé publique du destinataire :

$$c_K = E_{K_{\text{pub}}}(K_{\text{session}}).$$

Le message final envoyé comprend alors à la fois c_K et c . Le destinataire utilise sa clé privée pour déchiffrer K_{session} et, par la suite, récupère le message m .

Gestion des signatures avec GnuPG En plus du chiffrement, GnuPG permet de signer numériquement les messages. Le processus est le suivant :

1. Calcul du haché du message :

$$H(m),$$

où H est une fonction de hachage cryptographique (ex. SHA-256).

2. Signature numérique : Le haché est chiffré avec la clé privée pour générer la signature s :

$$s = \text{Sign}_{K_{\text{priv}}}(H(m)).$$

3. Vérification : Le destinataire déchiffre la signature avec la clé publique et compare le haché obtenu au calcul effectué sur le message reçu.

Formulations mathématiques et références Les mécanismes sous-jacents reposent sur des problèmes mathématiques difficiles, tels que la factorisation d'entiers ou le problème du logarithme discret. Pour davantage de détails mathématiques et techniques, nous nous référerons au manuel de GnuPG¹

ainsi qu'à d'autres ressources spécialisées en cryptographie.

L'utilisation de GnuPG dans notre application permet ainsi d'assurer la confidentialité, l'intégrité et l'authenticité des votes grâce à une combinaison judicieuse des techniques de chiffrement symétrique et asymétrique et du recours aux signatures numériques.

2.1.3 Transmission sécurisée des votes via emails

Dans notre application, la transmission des votes s'effectue par l'envoi de messages électroniques chiffrés, garantissant ainsi la confidentialité et l'intégrité des bulletins de vote. Pour ce faire, nous utilisons deux protocoles principaux : SMTP pour l'envoi et IMAP pour la réception des emails sécurisés.

Procédure d'envoi

1. **Préparation du message :** Chaque votant remplit un formulaire via l'interface web (Django) et génère un bulletin de vote. Ce bulletin, ainsi que son identité, sont chiffrés à l'aide de GnuPG, qui applique le chiffrement asymétrique (en utilisant la clé publique du centre de réception) et ajoute une signature numérique pour garantir l'intégrité du message.
2. **Chiffrement et signature :** Soit m le message contenant le bulletin de vote. La clé de chiffrement asymétrique permet de produire :

$$c = E_{K_{\text{pub}}}(m) \quad \text{et} \quad s = \text{Sign}_{K_{\text{priv}}}(H(m))$$

où $H(m)$ représente le haché du message, assurant ainsi que toute altération du contenu sera détectée.

3. **Envoi via SMTP :** Le message chiffré, accompagné de sa signature, est ensuite transmis via le protocole SMTP à l'adresse email dédiée du Centre de Comptage (CO) ou directement du Centre de Dépouillement (DE).

Procédure de réception

1. **Accès via IMAP :** Les centres CO et DE accèdent à leur boîte de réception en utilisant le protocole IMAP, ce qui leur permet de récupérer les emails de façon sécurisée.
2. **Extraction et traitement :** Les messages reçus sont extraits, stockés temporairement, puis traités pour vérifier la signature numérique et déchiffrer le contenu. Seule la clé privée correspondante permet de déchiffrer le message, garantissant ainsi que seul l'entité autorisée peut accéder au bulletin de vote.
3. **Validation :** Après déchiffrement, le système valide le vote et l'intègre dans le processus de dépouillement, tout en assurant que l'identité du votant reste dissociée de son choix.

¹<https://www.gnupg.org/gph/fr/manual.html#AEN189>

Comparaison des protocoles utilisés Le tableau ci-dessous résume les caractéristiques principales des protocoles SMTP et IMAP employés dans notre système :

Protocole	Fonction principale	Sécurité
SMTP	Envoi des emails	Utilisation de TLS pour sécuriser la connexion
IMAP	Réception et gestion des emails	Accès sécurisé via TLS/SSL

Table 3: Caractéristiques des protocoles de transmission des votes

Avantages de la transmission par emails chiffrés :

- **Confidentialité** : Même en cas d'interception, les messages restent illisibles sans la clé privée correspondante.
- **Traçabilité** : Chaque vote est transmis et peut être suivi depuis l'envoi jusqu'à la réception, permettant de vérifier l'intégrité du processus.
- **Fiabilité** : Le protocole IMAP assure une gestion efficace des emails, minimisant les risques de perte de messages.

Cette approche hybride, combinant le chiffrement des données via GnuPG et la robustesse des protocoles SMTP et IMAP, garantit une transmission sécurisée et fiable des votes dans le cadre de notre système de vote électronique.

2.2 Conception

2.2.1 Architecture globale du système

Le système de vote électronique repose sur une architecture sécurisée impliquant plusieurs entités interagissant par le biais de messages chiffrés. L'objectif est d'assurer l'authenticité, la confidentialité et l'intégrité des votes envoyés par les électeurs.

Principaux acteurs du système :

- **Votant (V)** : Génère son vote (B) et son identifiant (I), puis chiffre ces informations avant de les envoyer aux centres CO et DE.
- **Centre de Comptage (CO)** : Vérifie l'identité du votant et retransmet les informations au Centre de Dépouillement après un nouveau chiffrement.
- **Centre de Dépouillement (DE)** : Déchiffre et valide le vote en comparant les deux messages reçus.

Processus de transmission et validation des votes : Le processus se déroule comme suit :

1. Le votant chiffre son identifiant I avec la clé publique de CO ($K_p(CO)$) et son bulletin de vote B avec la clé publique de DE ($K_p(DE)$), puis envoie ces données au Centre CO.
2. Le Centre CO déchiffre I à l'aide de sa clé privée ($K_{pr}(CO)$), mais ne peut pas accéder à B . Il chiffre à nouveau (I, B) avec $K_p(DE)$ et transmet le message au Centre DE.
3. Le Centre DE déchiffre les deux messages (celui du votant et celui de CO) avec sa clé privée $K_{pr}(DE)$ et compare les deux. Si les informations correspondent, le vote est validé.

Schéma de l'architecture du système : L'architecture globale du système est illustrée par la figure suivante :

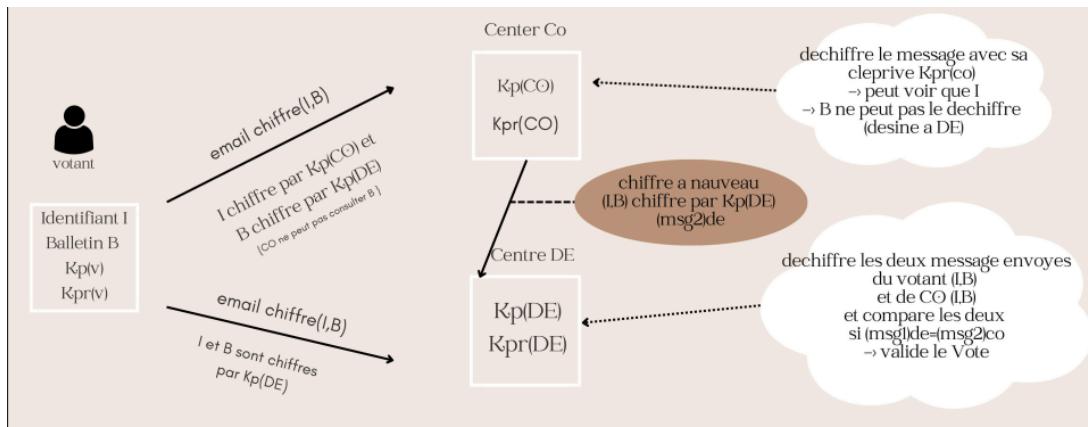


Figure 4: Schéma de l'architecture du système de vote sécurisé

Résumé des interactions des entités : Le tableau suivant résume les différentes interactions entre les acteurs du système :

Entité	Action	Chiffrement	Transmission
Votant (V)	Génération du vote (I, B)	I chiffré avec $K_p(CO)$, B avec $K_p(DE)$	Envoi à CO et DE
Centre CO	Vérification de I	Déchiffre I avec $K_{pr}(CO)$	Chiffre (I, B) avec $K_p(DE)$, transmet à DE
Centre DE	Validation du vote	Déchiffre I, B avec $K_{pr}(DE)$	Compare les deux messages et valide

Table 4: Résumé des interactions et traitements des entités du système

Cette architecture garantit que l'anonymat du votant est préservé tout en assurant l'intégrité et la vérifiabilité du scrutin.

2.2.2 Modélisation UML

Dans cette section, nous présentons la modélisation UML de notre système de vote électronique sécurisé. Pour illustrer le fonctionnement global de l'application, nous utilisons deux diagrammes clés : un diagramme de cas d'utilisation et un diagramme de séquence.

Diagramme de cas d'utilisation Ce diagramme décrit les interactions entre les acteurs principaux (le **Votant**, le **Centre CO** et le **Centre DE**) et le système de vote. Il met en évidence les étapes suivantes :

- Le Votant enregistre son vote et envoie des emails chiffrés contenant son identité et son bulletin.
- Le Centre CO reçoit l'email, déchiffre l'identité pour vérifier l'unicité du vote, puis re-chiffre les informations avant de les transmettre au Centre DE.
- Le Centre DE reçoit les emails du Votant et du Centre CO, déchiffre les messages, compare les informations et valide le vote.

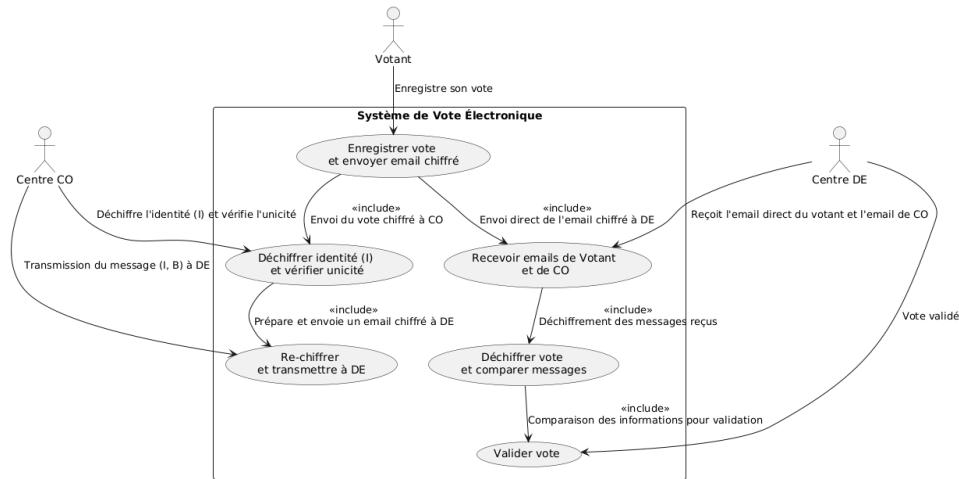


Figure 5: Diagramme de cas d'utilisation du système de vote

Diagramme de séquence Le diagramme de séquence illustre le flux d'interactions lors du processus de vote. Il détaille les échanges essentiels entre le Votant, le Centre CO et le Centre DE :

- Le Votant saisit et chiffre son vote, puis envoie deux emails chiffrés (l'un directement au Centre DE et l'autre au Centre CO).
- Le Centre CO déchiffre l'identité du votant, vérifie son unicité, re-chiffre les données, et envoie un email chiffré au Centre DE.
- Le Centre DE reçoit les deux emails, déchiffre les informations et compare les messages pour valider le vote.

Ces diagrammes, réalisés à l'aide de **PlantUML**, reflètent fidèlement la logique implémentée dans le code (voir fichier `views.py`) du code source et garantissent l'intégrité et l'anonymat du vote.

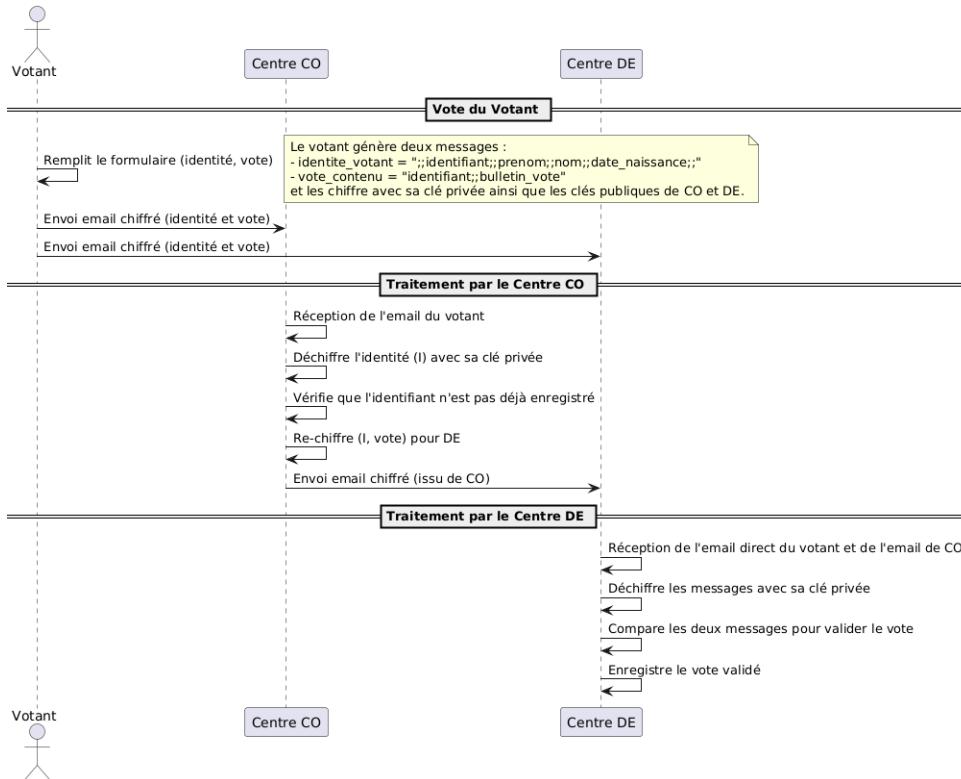


Figure 6: Diagramme de séquence du processus de vote électronique

2.2.3 Base de données et stockage sécurisé

La structure de la base de données a été conçue pour stocker de manière sécurisée les informations relatives aux votants, tout en préservant l'anonymat et la confidentialité des votes.

Modélisation des données Le projet utilise un modèle de données basé sur une classe abstraite `BaseVotant` dont héritent les classes `CoVotant` et `DeVotant`. Ces modèles permettent de stocker :

- **Les informations personnelles du votant** : nom, prénom, date de naissance et un identifiant unique.
- **Le bulletin de vote** : stocké sous forme chiffrée dans le champ `bulletinvote`.
- **La date du vote** : enregistrée automatiquement lors de la soumission.

Tableau récapitulatif des modèles

Sécurisation du stockage Pour garantir la sécurité des données :

- Le champ `identification` est défini comme unique et indexé, empêchant tout vote multiple.
- Le bulletin de vote est stocké sous forme chiffrée, de sorte que même en cas d'accès non autorisé à la base de données, le contenu reste illisible.

Modèle	Attributs principaux
BaseVotant	nom, prenom, datenaissance, identification, bulltinvote, date_vote
CoVotant	Hérite de BaseVotant (traitement des votes par le Centre CO)
DeVotant	Hérite de BaseVotant (stockage des votes validés par le Centre DE)

Table 5: Structure de la base de données pour le système de vote

- L'utilisation des modèles abstraits et de l'héritage permet une gestion claire et sécurisée des différentes étapes de traitement des votes (vérification par le Centre CO et validation par le Centre DE).

Cette approche de modélisation et de stockage garantit que toutes les données critiques sont protégées et que l'intégrité du vote est maintenue tout au long du processus.

2.3 Réalisation

2.3.1 Technologies et outils utilisés

Dans cette section, nous présentons les principales technologies et outils utilisés pour le développement de notre application de vote sécurisé. Chaque technologie joue un rôle spécifique dans l'architecture globale du système.

- **Django**


Rôle : Framework web côté *backend*. Django gère la logique métier, les vues, et l'interaction avec la base de données. Il offre également une sécurité intégrée et facilite le développement rapide d'applications web robustes.

- **SQLite3**



Rôle : Base de données utilisée pour stocker les informations des votants et les votes. SQLite3 est léger, simple à déployer et bien intégré à Django, assurant ainsi une gestion efficace et sécurisée des données.

- **GnuPG**



Rôle : Outil de chiffrement pour assurer la confidentialité et l'intégrité des votes. GnuPG implémente le chiffrement asymétrique ainsi que la signature numérique, garantissant que les messages (votes et identités) sont chiffrés et authentifiés lors de leur transmission.

- HTML, CSS & Bootstrap



Rôle : Technologies utilisées pour le développement de l'interface graphique (*front-end*). HTML structure le contenu, CSS le stylise, et Bootstrap offre un cadre responsive et moderne pour garantir une expérience utilisateur optimale sur tous types d'appareils.

2.3.2 Implémentation des fonctionnalités principales

Dans cette section, nous détaillons l'implémentation des fonctionnalités principales de notre système de vote électronique sécurisé. Le processus repose sur plusieurs étapes clés :

- La saisie et le chiffrement du vote par le votant.
- L'envoi de messages chiffrés vers le Centre CO et directement vers le Centre DE.
- Le traitement des emails reçus par le Centre CO (vérification et re-chiffrement) et leur transmission au Centre DE.

Fonction d'envoi d'email chiffré La fonction suivante permet d'envoyer un email contenant un message chiffré. Elle est utilisée par le votant pour transmettre ses informations de vote.

```
def send_encrypted_mail(to_email, subject, message):
    """Envoie un email avec un message chiffré."""
    email = EmailMessage(
        subject=subject,
        body=message,
        from_email=' ', # adresse du votant
        to=[to_email]
    )
    email.send()
```

Chiffrement et préparation des messages Avant d'envoyer les emails, le votant chiffre ses données. L'extrait de code ci-dessous montre comment sont chargées les clés publiques de CO et DE, la construction des messages à chiffrer, ainsi que leur chiffrement à l'aide de GnuPG :

```
# Charger les clés publiques de CO et DE
```

```

with open(os.path.join(GPG_KEYS_DIR, "pubkeyco.asc"), "r") as f:
    gpg.import_keys(f.read())
with open(os.path.join(GPG_KEYS_DIR, "pubkeyde.asc"), "r") as f:
    gpg.import_keys(f.read())

# Cr ation des messages  a chiffrer
identite_votant = f";{identifiant};{prenom};{nom};{date_naissance};""
vote_contenu = f"{identifiant};{bulletinvote}""

# Chiffrement avec les bonnes cl es publiques et signature avec la cl e priv e du
→ votant
identite_chiffree_co = gpg.encrypt(identite_votant, recipients=['@adresse CO'],
                                   sign=votant_private_fingerprint)
identite_chiffree_de = gpg.encrypt(identite_votant, recipients=['@adresse email
                                   → associe a DE'], sign=votant_private_fingerprint)

vote_chiffre_co = gpg.encrypt(vote_contenu, recipients=['@adresse email associe
                                   → a DE'], sign=votant_private_fingerprint)
vote_chiffre_de = gpg.encrypt(vote_contenu, recipients=['@adresse email associe
                                   → a DE'], sign=votant_private_fingerprint)

```

Envoi des emails vers CO et DE Une fois les messages chiffr s, ils sont envoy s aux centres correspondants :

```

send_encrypted_mail("@adresse CO", "votantiden_co", str(identite_chiffree_co))
send_encrypted_mail("@adresse CO", "votantres_co", str(vote_chiffre_co))
send_encrypted_mail("@adresse email associe a DE", "votantiden_de",
                    → str(identite_chiffree_de))
send_encrypted_mail("@adresse email associe a DE", "votantres_de",
                    → str(vote_chiffre_de))

messages.success(request, "Votre vote a  t   enregistr  avec succ s.")
return redirect('votant')

```

2.3.3 Gestion de la s curit 

La s curit  de notre application est assur e par plusieurs m canismes visant   garantir la confidentialit , l'int grit  et l'authenticit  des votes. Nous utilisons :

- **Le chiffrement asym trique via GnuPG** pour prot ger les donn es sensibles.

- La transmission sécurisée des emails à l'aide des protocoles SMTP (pour l'envoi) et IMAP (pour la réception).
- La vérification des signatures numériques afin de s'assurer de l'authenticité des messages.

Accès aux emails via IMAP La fonction suivante récupère les emails chiffrés depuis la boîte de réception, garantissant que seuls les messages authentiques et complets sont traités.

```
def receive_encrypted_mail(user, password, save_directory, nbrmessage):
    try:
        mail = imaplib.IMAP4_SSL("imap.gmail.com")
        mail.login(user, password)
        mail.select("inbox")
        result, data = mail.search(None, "ALL")
        email_ids = data[0].split()
        for e_id in email_ids[-nbrmessage:]:
            result, msg_data = mail.fetch(e_id, "(RFC822)")
            raw_email = msg_data[0][1]
            msg = email.message_from_bytes(raw_email, policy=default)
            subject = msg["subject"]
            content = ""
            if msg.is_multipart():
                for part in msg.walk():
                    if part.get_content_type() == "text/plain":
                        content += part.get_payload(decode=True).decode()
            else:
                content = msg.get_payload(decode=True).decode()
    except Exception as e:
        print(f"Erreur lors de la réception des emails : {str(e)}")
```

Déchiffrement et vérification des messages Pour assurer l'intégrité des votes, la fonction suivante déchiffre les fichiers reçus et vérifie la validité de la signature :

```
def decrypt_and_verify_file(privkey, pubkey, filedcry, output):
    gpg = gnupg.GPG(gpgbinary="C:/Program Files (x86)/gnupg/bin/gpg.exe")
    privateKeyFile = os.path.join(GPG_KEYS_DIR, privkey)
    with open(privateKeyFile, "r") as f:
        gpg.import_keys(f.read())
    file_path = os.path.join("C:/Users/oussa/voting_system/co", filedcry)
    print(f"Déchiffrement du fichier : {file_path}")
```

```

with open(file_path, "rb") as f:
    decrypted_data = gpg.decrypt_file(f,
        ↳ output=os.path.join("C:/Users/oussa/voting_system/co", output))
return decrypted_data.ok

```

Envoi d'emails du Centre CO vers le Centre DE Pour la transmission des votes validés, le Centre CO envoie un email chiffré vers le Centre DE en utilisant la fonction suivante :

```

def send_encrypted_mail_co(to_email, subject, message):
    """
    Envoie un email chiffré depuis le compte CO.
    """

    connection = get_connection(
        backend='django.core.mail.backends.smtp.EmailBackend',
        host='smtp.gmail.com',
        port=587,
        username='@adresse CO', # adresse du Centre CO
        password='APP PASSWORD adresse CO ',
        use_tls=True,
    )
    email = EmailMessage(
        subject=subject,
        body=message,
        from_email='@adresse CO',
        to=[to_email],
    )
    email.send(fail_silently=False)

```

Résumé et explications

- **Chiffrement et signatures** : Les données du votant (identité et bulletin) sont chiffrées avec GnuPG en utilisant les clés publiques des centres CO et DE, et signées avec la clé privée du votant. Cela garantit que seul le destinataire autorisé peut déchiffrer le message et que toute altération est détectée.
- **Transmission sécurisée** : Les messages chiffrés sont envoyés via SMTP, et leur réception est assurée par IMAP. Le déchiffrement par CO et DE est effectué avec leurs clés privées respectives.
- **Gestion de la sécurité** : La combinaison du chiffrement asymétrique, des signatures numériques et de la gestion sécurisée des emails assure l'intégrité et la confidentialité des votes tout au long du processus.

3 Résultats et manuel d'utilisation

3.1 Interface graphique d'un votant

L'interface du votant permet aux utilisateurs de soumettre leur vote de manière sécurisée. Elle comprend :

- Un champ pour saisir son identifiant afin de s'authentifier.
- Une liste des candidats parmi lesquels il peut faire son choix.
- Un bouton permettant de valider et chiffrer son vote avant son envoi.

Une fois le vote soumis, le votant reçoit une confirmation indiquant que son vote a été pris en compte. La figure 8 illustre cette interface.

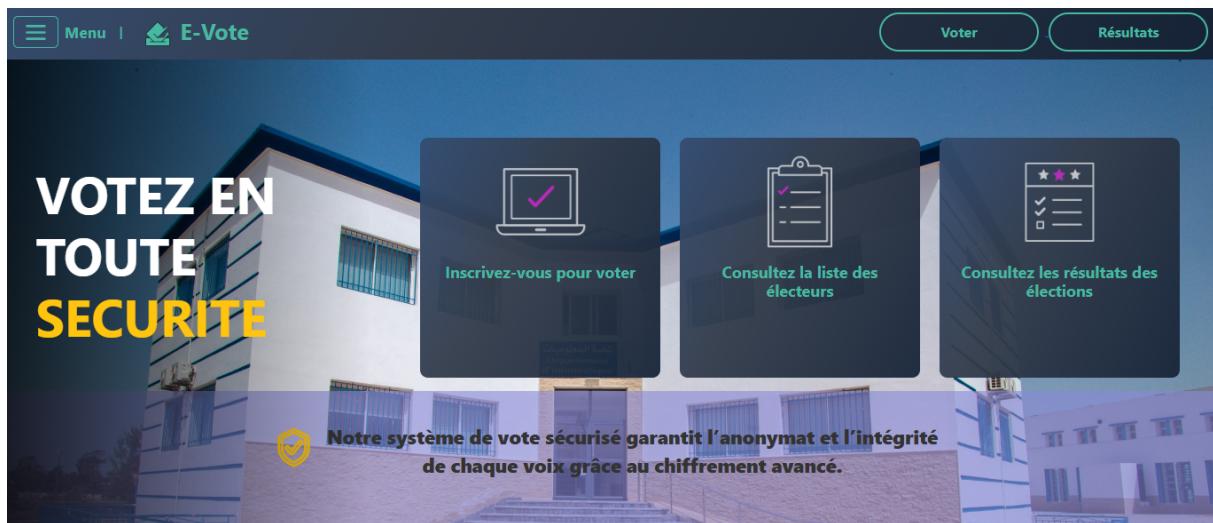


Figure 7: Interface d'accueil



Figure 8: Interface graphique du votant

3.2 Interface graphique du centre de comptage (CO)

Le centre de comptage (CO) est responsable de vérifier l'identité des votants et de transmettre les votes chiffrés au centre de dépouillement (DE). L'interface comprend :

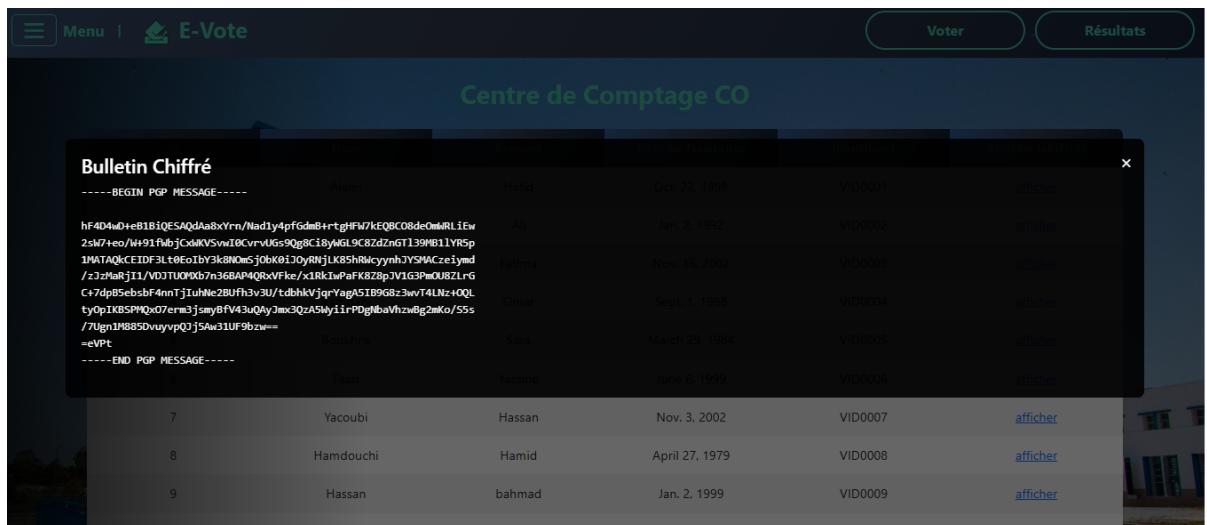
- Une liste des votes reçus avec l'identifiant du votant et l'état de vérification.
- Un bouton permettant de valider ou rejeter un vote en fonction des critères d'authentification.
- Une option pour transmettre les votes vérifiés au centre de dépouillement.
- Affichage de bulletin de vote chiffré.

La figure 10 montre l'interface du centre de comptage.



ID	Nom	Prénom	Date de Naissance	Identifiant	Bulletin (chiffré)
1	Alami	Hafid	Oct. 22. 1998	VID0001	afficher
2	Bariki	Ali	Jan. 2, 1992	VID0002	afficher
3	El Mansouri	Fatima	Nov. 16. 2002	VID0003	afficher
4	Ramdani	Omar	Sept. 1, 1988	VID0004	afficher
5	Boukhris	Sara	March 29, 1984	VID0005	afficher
6	Fassi	Yassine	June 6, 1999	VID0006	afficher
7	Yacoubi	Hassan	Nov. 3, 2002	VID0007	afficher
8	Hamdouchi	Hamid	April 27, 1979	VID0008	afficher
9	Hassan	bahmad	Jan. 2, 1999	VID0009	afficher

Figure 9: Interface graphique du centre de comptage (CO)



ID	Nom	Prénom	Date de Naissance	Identifiant	Bulletin (chiffré)
1	Alami	Hafid	Oct. 22. 1998	VID0001	afficher
2	Bariki	Ali	Jan. 2, 1992	VID0002	afficher
3	El Mansouri	Fatima	Nov. 16. 2002	VID0003	afficher
4	Ramdani	Omar	Sept. 1, 1988	VID0004	afficher
5	Boukhris	Sara	March 29, 1984	VID0005	afficher
6	Fassi	Yassine	June 6, 1999	VID0006	afficher
7	Yacoubi	Hassan	Nov. 3, 2002	VID0007	afficher
8	Hamdouchi	Hamid	April 27, 1979	VID0008	afficher
9	Hassan	bahmad	Jan. 2, 1999	VID0009	afficher

Bulletin Chiffré

```
--BEGIN PGP MESSAGE-- Alami Hafid Oct. 22. 1998 VID0001 afficher
hF4D4wH+eB1BiQESAOAa8xYrn/Nad1y4pfGdmB+rtgHF7kEOBC08deOmkRL1Ew
2sw7+eo/W+91fWbjCxWKVSvn10CvrvUgs9Qg0Ci8yWGl9C8ZdZngT139NB11YRp
1MATA0kCEIDf3L+t0Eo1by3k8Wm5jcbk0i0Jy0yRNjK85hRwcyhhJYSMACzeiymd
/z3zhArJ11/WDJTUM0K67n368AP4Q0bxFke/x1k1nPwFAK82pJViG3PmOU82LrG
C+7dp85eb5b4mNtjiuhne2BUfh3v3U/tdbhkVjqrYagAS19Gbz3wT4LNz+QQL
tyOpTKBSPM0Qd7ern3jsayBfv43uQhyJmx3QzASMyiir+PdgibaH:zwBg2mKo/s5s
/7lgn1H885DvuyvpQ2j5Aw31UF9bzw==
--evPt
-----END PGP MESSAGE----- Fassi Yassine June 6, 1999 VID0006 afficher

```

Figure 10: Affichage de bulletin de vote chiffré

3.3 Interface graphique du centre de dépouillement (DE) et Résultats

Le centre de dépouillement (DE) est chargé de déchiffrer et de traiter les votes validés par le centre de comptage. L'interface comprend :

- Une liste des votes reçus avec leur statut.
- Un bouton permettant de démarrer le processus de dépouillement.

Une fois le dépouillement terminé, les résultats du vote sont affichés sous forme de tableau ou de graphique. La figure 11 montre l'interface du centre de dépouillement, et la figure 12 affiche les résultats finaux.



The screenshot shows a web-based application for election tallying. At the top, there's a header with 'Menu' and 'E-Vote' on the left, and 'Voter' and 'Résultats' on the right. Below the header, the title 'Centre de Dépouillement DE' is centered. A table follows, with columns labeled 'ID', 'Nom', 'Prénom', 'Date de Naissance', 'Identifiant', and 'Bulletin (déchiffré)'. The table contains 9 rows of data, each representing a voter with their ID, name, first name, birth date, identifier, and decrypted ballot status.

ID	Nom	Prénom	Date de Naissance	Identifiant	Bulletin (déchiffré)
1	Alami	Hafid	Oct. 22, 1998	VID0001	candidat1
2	Bariki	Ali	Jan. 2, 1992	VID0002	candidat1
3	El Mansouri	Fatima	Nov. 16, 2002	VID0003	candidat2
4	Ramdani	Omar	Sept. 1, 1988	VID0004	candidat3
5	Boukhris	Sara	March 29, 1984	VID0005	candidat3
6	Fassi	Yassine	June 6, 1999	VID0006	candidat3
7	Yacoubi	Hassan	Nov. 3, 2002	VID0007	candidat4
8	Hamdouchi	Hamid	April 27, 1979	VID0008	candidat5
9	Hassan	bahmad	Jan. 2, 1999	VID0009	candidat5

Figure 11: Interface graphique du centre de dépouillement (DE)

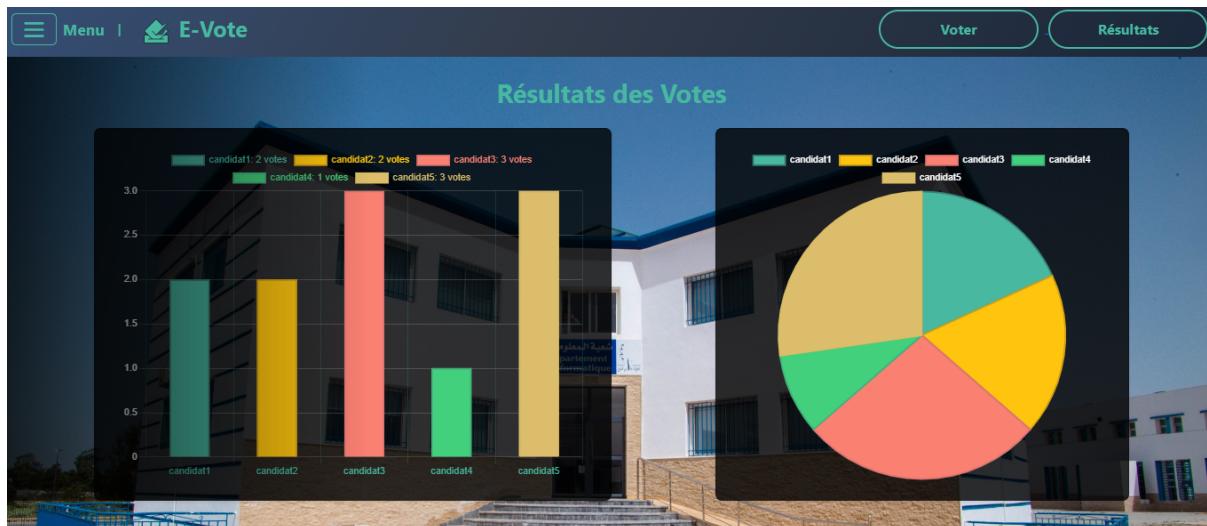


Figure 12: Page des résultats après le dépouillement

3.4 Envoi des emails

L'un des aspects essentiels du système de vote électronique est la transmission sécurisée des votes via email. Une fois que le votant a rempli son bulletin de vote, celui-ci est chiffré en utilisant la clé publique du Centre de Comptage (CO) et du Centre de Dépouillement (DE). Ce processus garantit que seul le destinataire prévu peut déchiffrer et traiter le vote.

Dans notre implémentation, l'application envoie deux emails distincts :

- Un premier email est envoyé au Centre de Comptage (CO) contenant les informations du votant et son vote chiffré.
- Un second email est adressé au Centre de Dépouillement (DE) avec une copie du vote chiffré pour validation et comptabilisation.

Les images suivantes illustrent l'envoi des emails par l'application :

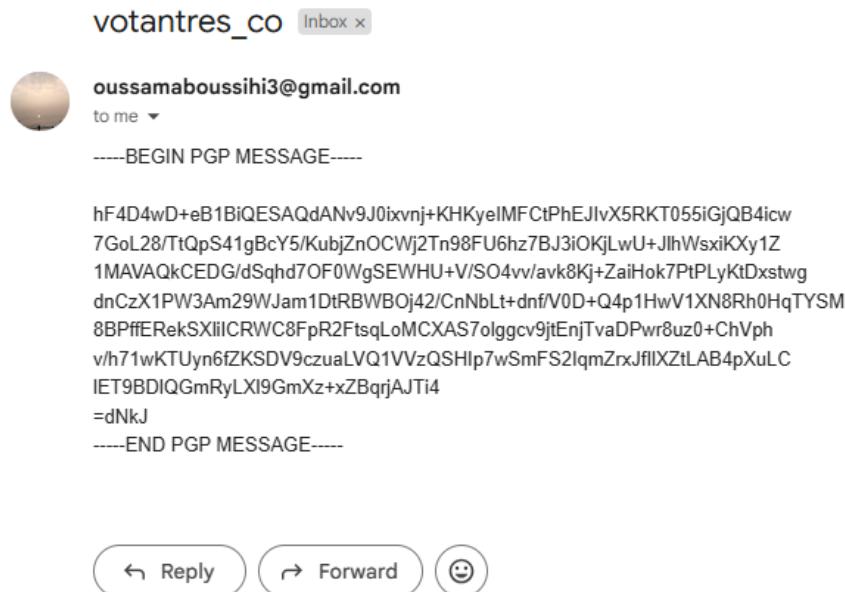


Figure 13: Envoi de l'email chiffré au Centre de Comptage (CO)

Grâce à ce mécanisme, chaque vote est acheminé de manière sécurisée vers les centres responsables, empêchant toute modification ou interception non autorisée. L'intégrité et la confidentialité des votes sont ainsi garanties tout au long du processus électoral.



Figure 14: Envoi de l'email chiffré au Centre de Dépouillement (DE)

4 Conclusion

4.1 Évaluation des résultats obtenus

Ce projet a permis de concevoir et de développer une application de vote électronique sécurisée en utilisant le chiffrement asymétrique OpenPGP. Grâce à l'architecture mise en place, nous avons pu garantir les principes fondamentaux d'un vote électronique fiable, notamment :

- **Confidentialité** : Les votes sont chiffrés avant leur transmission, empêchant toute divulgation non autorisée.
- **Authentification** : Chaque votant est identifié avant de soumettre son vote, évitant ainsi les fraudes électorales.
- **Intégrité** : Une double vérification entre le Centre de Comptage (CO) et le Centre de Dépouillement (DE) assure que chaque vote reçu est valide et n'a pas été modifié.
- **Anonymat** : Le Centre de Comptage (CO) ne peut pas accéder au contenu des votes, tandis que le Centre de Dépouillement (DE) ne connaît pas l'identité des votants.
- **Fiabilité** : L'utilisation d'algorithmes de chiffrement robustes empêche toute altération des données pendant le processus électoral.

L'application a été testée avec succès dans un environnement contrôlé, confirmant que les messages chiffrés sont bien transmis et traités entre les différentes entités du système. De plus, l'intégration avec GnuPG s'est révélée efficace pour le chiffrement et le déchiffrement des votes.

Néanmoins, bien que le système ait atteint ses objectifs de sécurité et de transparence, certaines améliorations peuvent encore être envisagées pour renforcer l'expérience utilisateur et la performance globale.

4.2 Perspectives d'amélioration

Plusieurs pistes d'amélioration peuvent être explorées afin d'optimiser notre système de vote électronique :

- **Mise en place d'un système d'authentification sécurisé** : Actuellement, le système permet aux votants, au Centre de Comptage (CO) et au Centre de Dépouillement (DE) de traiter les votes de manière sécurisée. Toutefois, une authentification forte (via mot de passe chiffré, OTP ou authentification biométrique) garantirait une identification fiable des utilisateurs avant d'accéder aux fonctionnalités critiques.
- **Intégration d'une interface administrateur** : Une interface dédiée à un administrateur pourrait être ajoutée pour gérer l'ensemble du processus électoral, notamment la création et la gestion des votants, la configuration des clés de chiffrement, la surveillance du scrutin et la validation des résultats.
- **Extension à une architecture distribuée avec blockchain** : Actuellement basé sur une structure centralisée, le système pourrait évoluer vers une architecture distribuée utilisant la blockchain pour garantir la transparence, l'immutabilité et la traçabilité totale des votes, réduisant ainsi les risques de fraude ou de manipulation.
- **Audit de sécurité et certification** : Un audit de sécurité approfondi pourrait être mené pour identifier d'éventuelles vulnérabilités et renforcer la robustesse du système. Une certification par des experts en cybersécurité augmenterait la confiance des utilisateurs et des institutions dans ce mode de vote électronique.
- **Gestion avancée des erreurs et des tentatives de fraude** : La mise en place de mécanismes de détection d'anomalies et de logs détaillés permettrait d'identifier et d'alerter sur d'éventuelles tentatives de fraude ou de dysfonctionnements lors du scrutin.
- **Amélioration de la compatibilité et de l'accessibilité** : Adapter l'application pour une utilisation sur divers appareils (smartphones, tablettes, navigateurs web) garantirait une accessibilité optimale aux électeurs, réduisant ainsi les barrières techniques à la participation au vote.

En conclusion, ce projet a démontré la faisabilité d'un système de vote électronique sécurisé basé sur OpenPGP. Les résultats obtenus sont encourageants, et plusieurs axes d'amélioration permettraient de renforcer encore davantage la sûreté, l'accessibilité et la fiabilité du système, ouvrant la voie à une adoption à plus grande échelle dans des contextes réels.

En conclusion, ce projet a démontré la faisabilité d'un système de vote électronique sécurisé reposant sur OpenPGP. Les résultats obtenus sont encourageants, et les améliorations futures pourraient renforcer encore davantage la sûreté et l'accessibilité du système, ouvrant la voie à une adoption à plus grande échelle dans des contextes réels.

5 Bibliographie

- **GnuPG Manual:** Documentation officielle de GnuPG. Disponible en ligne :
<https://gnupg.org/documentation/manuals.html>
- **Django Documentation:** Documentation officielle du framework Django. Disponible en ligne : <https://docs.djangoproject.com/en/5.1/>
- **Bootstrap Documentation:** Documentation officielle du framework Bootstrap. Disponible en ligne : <https://getbootstrap.com/docs/5.3/getting-started/introduction/>
- **GPGVote:** Un système de vote basé sur GPG. Repository GitHub :
<https://github.com/Ernest0x/gpgvote>
- **Decentralized Voting System:** Un projet de vote décentralisé. Repository GitHub : <https://github.com/Krish-Depani/Decentralized-Voting-System>
- **Cryptographie et Sécurité Informatique :** Pour une compréhension approfondie des algorithmes de cryptographie modernes, voir le livre de Bruce Schneier *Applied Cryptography*, ou la documentation de la bibliothèque OpenPGP :
<https://www.openpgp.org/resources/>
- **Systèmes de Vote Électronique:** Un aperçu des techniques de vote sécurisé :
<https://www.usenix.org/conference/evtvote20/presentation>