S

LIBERTY Shield

Whitepaper sécurité

Document de référence technique. Décrit l'architecture cryptographique réellement déployée, le modèle de menace et les mesures de protection.

À jour au 26 juin 2026.

Liberty Club LLC — Sheridan, Wyoming, USA — contact@libertyclub.finance

1. Principe

Liberty Shield est un coffre-fort numérique **zero-knowledge**. Tout le chiffrement et le déchiffrement ont lieu **sur l'appareil de l'utilisateur**, dans le navigateur. Le serveur ne reçoit, ne stocke et ne transmet jamais que des données déjà chiffrées, opaques. Il ne détient aucune clé, et ne connaît jamais le mot de passe maître. Conséquence assumée : si l'utilisateur perd ses accès, personne — pas même nous, pas même sur injonction — ne peut les récupérer à sa place.

Cette garantie repose sur des primitives cryptographiques standard, fournies par **libsodium** (aucune cryptographie « maison »), et sur une hiérarchie de clés vérifiable décrite ci-dessous.

---

2. Architecture cryptographique

2.1 Primitives

Fonction	Algorithme	Paramètres
Chiffrement des données	XChaCha20-Poly1305 (AEAD)	nonce aléatoire de 24 octets, généré à chaque opération
Dérivation de clé (mot de passe → clé)	Argon2id	64 Mio de mémoire, 3 itérations, sortie 256 bits, sel aléatoire de 16 octets
Empreinte d'authentification	BLAKE2b (avec séparation de domaine)	domaine `shield:auth:v1`
Partage / messages scellés	X25519 (`crypto_box_seal`)	scellé anonyme vers la clé publique du destinataire
Transmission (héritage)	SLIP-39 (partage de secret de Shamir, à seuil)	seuil M-sur-N configurable
Stockage serveur de l'empreinte d'auth	scrypt + sel par compte	N=16384, r=8, p=1

Le chiffrement est **authentifié** (AEAD) : toute altération d'un message chiffré est détectée et provoque un échec de déchiffrement. Le nonce de 24 octets de XChaCha20-Poly1305 est tiré aléatoirement à chaque chiffrement ; sa taille étendue rend toute collision négligeable et évite la classe de failles liée à la réutilisation de nonce.

2.2 Hiérarchie de clés

1. Le **mot de passe maître** est transformé en **MasterKey** par Argon2id (jamais transmis au serveur).
2. La MasterKey enveloppe une **AccountKey** aléatoire (256 bits) — la racine du compte.

3. L'AccountKey enveloppe la **clé privée X25519** de l'utilisateur (pour recevoir des éléments partagés ou un legs).
4. Chaque coffre possède une **VaultKey** aléatoire, enveloppée par l'AccountKey.
5. Chaque fiche possède sa **propre clé** (ItemKey), enveloppée par la VaultKey.

Ce modèle « une clé par fiche » permet de partager un élément précis sans exposer le reste du coffre, et de révoquer un accès sans tout rechiffrer. Le changement de mot de passe maître ne ré-enveloppe que l'AccountKey : aucune fiche n'est rechiffrée.

2.3 Ce que le serveur voit — et ne voit jamais

Reçu et stocké par le serveur : des enveloppes chiffrées opaques (base64), une empreinte d'authentification (elle-même re-hachée par scrypt avec un sel propre au compte, puis comparée en **temps constant**), les paramètres publics de dérivation (sel, mémoire, itérations), et la clé **publique** de l'utilisateur.

Jamais reçu : le mot de passe maître, la MasterKey, l'AccountKey, les VaultKeys, les ItemKeys, la clé privée, le contenu des fiches en clair. Une fuite intégrale de la base de données ne révèle donc ni les mots de passe, ni le contenu des coffres.

---

3. Les niveaux de sensibilité

Shield organise les secrets en niveaux, chacun avec son propre verrou.

- **Courant** — déverrouillé par le mot de passe maître seul. Usage quotidien (identifiants, cartes, codes wifi, abonnements).
- **Confidentiel** — exige le mot de passe maître **et** un code à six chiffres. La clé de ce niveau combine l'AccountKey et le code : le code seul ne suffit pas, et le mot de passe maître seul non plus. Ce code rapide protège avant tout contre un accès opportuniste à une session déjà ouverte ; les codes trivialement faibles (suites, répétitions, codes courants) sont **refusés à la création**. Pour les secrets réellement critiques, c'est le Sanctuaire — verrouillé par une phrase à haute entropie — qui constitue le rempart cryptographique.
- **Sanctuaire** — exige le mot de passe maître **et** une phrase secrète. Destiné aux secrets les plus critiques (seeds de wallets). La phrase, à haute entropie, place ce niveau hors d'atteinte d'un brute-force même si le mot de passe maître est compromis.

3.1 La feuille de secours

À l'inscription, une feuille de secours unique est générée : 160 bits d'aléa, formatés en groupes lisibles, à imprimer et conserver hors-ligne. Elle enveloppe une seconde fois l'AccountKey, permettant de **récupérer un mot de passe maître oublié**. Son entropie la rend mathématiquement imbruteforçable ; son seul risque est physique (perte ou vol du papier). Elle est régénérable à tout moment (la nouvelle invalide l'ancienne) et n'est jamais stockée en clair côté serveur.

---

4. Transmission / héritage

Le legs ne repose sur aucune confiance dans l'éditeur : il repose sur les mathématiques.

1. Une **clé de legs** aléatoire (256 bits) chiffre le coffre transmis (le *bundle*).
2. Cette clé est découpée en **fragments SLIP-39** selon un seuil M-sur-N choisi par le titulaire.
3. Les fragments sont remis **en main propre** aux porteurs désignés (bénéficiaires, exécuteurs).
4. Le coffre chiffré est conservé ; **la clé ne l'est jamais** — elle n'existe que recomposée à partir du seuil de fragments.
5. Les héritiers recomposent la clé et ouvrent le coffre via un **décrypteur hors-ligne** (fichier statique autonome), **sans aucune dépendance à nos serveurs** : la récupération fonctionne même si Shield, l'entreprise, n'existe plus.

4.1 Machine à états du déclenchement

Le déclenchement est gouverné par une machine à états automatisée (vérification horaire) :

`ARMED` → `PING_OVERDUE` → `PENDING_RELEASE` → `RELEASED`

- **ARMED** : signal de vie actif. Le titulaire confirme sa présence à intervalle régulier (`livenessIntervalDays`).
- **PING_OVERDUE** : un signal de vie a été manqué ; un email de rappel est envoyé. Un délai de grâce (`graceDays`) s'applique.
- **PENDING_RELEASE** : faute de réponse après le délai de grâce — ou sur demande des porteurs atteignant le seuil — une fenêtre de refus (`refusalWindowDays`) s'ouvre. Le titulaire est notifié et peut **annuler à tout moment**.
- **RELEASED** : à la fin de la fenêtre, le legs devient disponible aux porteurs. L'ouverture exige toujours la réunion des fragments physiques.

Le titulaire garde le contrôle à chaque étape : répondre au signal de vie réarme le système et annule tout déclenchement. Le réglage de ces délais (signal de vie, grâce, fenêtre de refus) se fait dans le module Héritage, bloc « Surveillance ».

4.2 Kit de secours hors-ligne et double accès

À la composition du legs, le titulaire peut télécharger deux fichiers : le **coffre chiffré** (`.shield`, illisible sans les fragments) et le **décrypteur** (page autonome). Archivés ensemble — ou confiés à l'exécuteur — ils forment un kit de secours qui garantit l'ouverture du legs **même si Shield, le service, disparaissait**. Le jour venu, deux chemins coexistent : si le service est en ligne, les héritiers retéléchargent eux-mêmes le coffre depuis leur accès une fois le legs libéré ; sinon, ils utilisent le kit archivé. Dans les deux cas, l'ouverture exige la réunion du seuil de fragments physiques. Le décrypteur est par ailleurs disponible en permanence comme fichier public, sans compte ni connexion.

5. Coffre familial

Compartiment partagé pour les secrets communs (wifi, abonnements, codes). La VaultKey du coffre familial est enveloppée par l'AccountKey du gérant, puis partagée aux membres via scellé X25519 (chaque membre déchiffre avec sa propre clé privée). Les droits d'édition sont gérés par membre. L'éditeur ne peut jamais lire le contenu.

---

6. Modèle de menace

Le point essentiel, et contre-intuitif : **le chiffrement n'est pas le maillon faible.** Les primitives employées sont à l'état de l'art et alignées sur les meilleurs acteurs audités du marché. Les risques réels se situent ailleurs.

1. **La force du secret choisi par l'utilisateur.** C'est le seul mur réellement franchissable par force brute (voir §7). Mesure **en place** : à l'inscription, longueur minimale imposée, jauge de robustesse, rejet des mots de passe triviaux, et encouragement explicite d'une phrase de plusieurs mots.
2. **L'appareil de l'utilisateur.** Un logiciel malveillant (keylogger, vol de presse-papier) peut capter le mot de passe **avant** que le chiffrement ne s'applique. Aucun chiffrement ne protège de ce cas, par construction : la protection porte sur les données stockées et transmises, pas sur la frappe en clair. Mesure : hygiène de l'appareil, déverrouillage rapide pour limiter les saisies, sensibilisation.
3. **La livraison du code.** Comme tout coffre opérant dans un navigateur, le code qui chiffre est servi par le serveur ; un code remplacé pourrait capter le secret. Mesures **en place** : Content-Security-Policy stricte (`default-src 'self'`, `connect-src 'self'`) qui empêche tout script d'exfiltrer des données vers un domaine tiers ; service worker qui met le code en cache après installation (l'application installée ne re-télécharge pas son code à chaque visite) ; **module de chiffrement client et décrypteur publiés en open-source**, vérifiables par quiconque. Renfort possible à terme : build reproductible avec empreintes publiées, et extension navigateur signée pour l'usage quotidien. Le SRI n'apporterait rien ici : les fichiers sont servis en même origine, sans CDN tiers.
4. **Les bugs d'implémentation.** Vérifiés par revue de code et, à terme, par audit tiers (voir §9).

---

7. Coût d'une attaque par force brute

Hypothèses : attaquant très bien doté (~1 000 GPU haut de gamme). Argon2id à 64 Mio / 3 itérations limite l'attaque à quelques centaines de milliers d'essais par seconde, chaque essai étant coûteux en mémoire. Le chiffrement rend chaque tentative chère ; c'est l'entropie du secret qui fixe le nombre de tentatives.

Secret choisi	Entropie	Temps de cassage estimé
Mot de passe faible (8 lettres)	~38 bits	quelques jours
Phrase de 4 mots	~52 bits	plusieurs siècles
Phrase de 5 mots	~64 bits	hors échelle humaine
Feuille de secours	160 bits	mathématiquement hors d'atteinte

Par niveau : au **Courant**, le mur est le mot de passe maître. Au **Confidentiel**, un second secret s'ajoute — un attaquant sans le mot de passe maître doit casser les deux ; avec le mot de passe maître, la sécurité dépend de la qualité du code. Au **Sanctuaire**, la phrase additionnelle place le niveau hors d'atteinte quoi qu'il arrive.

Économie de l'attaque. Pour un compte individuel, le coût d'un brute-force dépasse de loin la valeur du contenu : l'attaque par force n'est jamais le scénario rationnel. Un attaquant ira vers le moins cher — hameçonnage, logiciel malveillant, code piégé — d'où la priorité donnée aux défenses du §6 plutôt qu'au chiffrement, déjà imprenable.

---

8. Positionnement face au marché

Primitives vérifiées chez les concurrents de référence :

Capacité	Shield	Bitwarden	Proton Pass	1Password	Vault12	Inheriti
Chiffrement authentifié moderne	XChaCha20	✓	AES-GCM	AES-GCM	✓	AES-256
KDF Argon2id	✓	option	✓	— (PBKDF2)	n/a	n/a
Une clé par fiche	✓	—	✓	—	n/a	n/a
Zero-knowledge	✓	✓	✓	✓	✓	✓
Niveaux de sensibilité gradués	✓	—	—	—	—	—
Coffre familial partagé	✓	✓	✓	✓	—	—
Seeds crypto en type natif	✓	—	—	—	✓	✓
Transmission par seuil (M-sur-N)	✓	—	—	—	✓	✓
Récupération héritiers hors-ligne	✓	—	—	—	✓	✓
Tout-en-un (mdp + docs + seeds + héritage)	✓	—	—	—	—	—
Audit tiers indépendant	à venir	✓	✓	✓	~	✓

Shield emploie les mêmes briques modernes que les meilleurs (XChaCha20-Poly1305 comme NordPass, Argon2id et clé-par-fiche comme Proton Pass), et la fondation à seuil des spécialistes de l'héritage (Vault12, Inheriti) — mais intégrée à un coffre complet plutôt que réservée aux seeds.

---

9. Mesures en place et feuille de route

Déjà en place : chiffrement libsodium état de l'art ; Argon2id 64 Mio / 3 ; zero-knowledge vérifié ; empreinte d'auth re-hachée scrypt + comparaison à temps constant ; Content-Security-Policy stricte (anti-exfiltration) ; **limitation des tentatives de connexion** (rate-limiting sur connexion, pré-connexion et inscription) ; **protection contre l'énumération des comptes** ; **jauge de robustesse et rejet des mots de passe triviaux à l'inscription** ; **rejet des codes Confidentiel trivialement faibles** ; consentement horodaté à l'inscription ; suppression de compte conforme RGPD ; décrypteur hors-ligne autonome ; **module de chiffrement client et décrypteur publiés en open-source** (audit communautaire, transparence).

Feuille de route : build reproductible avec empreintes de build publiées ; extension navigateur signée pour l'usage quotidien ; programme de divulgation responsable des vulnérabilités ; audit de sécurité tiers indépendant lorsque la base de membres le justifiera.

Liberty Club LLC — Sheridan, Wyoming, USA. contact@libertyclub.finance