```
/*!
 * fp-altcha.js -- kleiner, eigenstaendiger Loeser fuer das klassische
ALTCHA V1
 * Proof-of-Work-Verfahren (SHA-256 Hashcash-Stil).
 *
 * Sucht fuer eine vom Server vorgegebene Aufgabe {algorithm, challenge,
salt, maxnumber, signature}
 * eine Zahl n mit 0 <= n <= maxnumber, fuer die SHA-256(salt + n) ==
challenge gilt, und traegt das
 * Ergebnis base64-kodiert in ein verstecktes Formularfeld "altcha" ein. Die
Verifikation auf dem
 * Server erfolgt mit der offiziellen ALTCHA-PHP-Bibliothek (siehe
src/Vendor/AltchaOrg/Altcha/V1).
 *
 * Laeuft vollstaendig lokal im Browser -- es wird keine Anfrage an einen
Drittanbieter gesendet.
 */
(function () {
    'use strict';

    function bufferToHex(buffer) {
        var bytes = new Uint8Array(buffer);
        var hex = '';
        for (var i = 0; i < bytes.length; i++) {
            var h = bytes[i].toString(16);
            hex += h.length === 1 ? '0' + h : h;
        }
        return hex;
    }

    function base64Encode(str) {
        var bytes = new TextEncoder().encode(str);
        var binary = '';
        for (var i = 0; i < bytes.length; i++) {
            binary += String.fromCharCode(bytes[i]);
        }
        return btoa(binary);
    }

    async function solve(algorithm, salt, target, maxNumber) {
        if (algorithm !== 'SHA-256') {
            // Nur SHA-256 wird von diesem kleinen Loeser unterstuetzt.
            return null;
        }

        var encoder = new TextEncoder();

        for (var n = 0; n <= maxNumber; n++) {
            var data = encoder.encode(salt + n);
            var digest = await crypto.subtle.digest('SHA-256', data);
```

```
            if (bufferToHex(digest) === target) {
                return n;
            }
        }

        return null;
    }

    async function processWidget(container) {
        var raw = container.getAttribute('data-fp-altcha');
        if (!raw) {
            return;
        }

        var task;
        try {
            task = JSON.parse(raw);
        } catch (e) {
            return;
        }

        var status = container.querySelector('.fp-altcha-status');
        var input = container.querySelector('.fp-altcha-input');
        if (!input) {
            return;
        }

        try {
            var number = await solve(task.algorithm, task.salt,
task.challenge, task.maxnumber);

            if (number === null) {
                if (status) {
                    status.textContent = container.getAttribute('data-label-
failed') || 'Sicherheitspruefung fehlgeschlagen. Bitte Seite neu laden.';
                }
                container.setAttribute('data-fp-altcha-state', 'failed');
                return;
            }

            var payload = {
                algorithm: task.algorithm,
                challenge: task.challenge,
                number: number,
                salt: task.salt,
                signature: task.signature
            };

            input.value = base64Encode(JSON.stringify(payload));
```

```
            container.setAttribute('data-fp-altcha-state', 'verified');

            if (status) {
                status.innerHTML = '<i class="fa fa-check" aria-
hidden="true"></i> ' +
                    (container.getAttribute('data-label-verified') ||
'Sicherheitspruefung bestaetigt');
            }
        } catch (e) {
            container.setAttribute('data-fp-altcha-state', 'failed');
            if (status) {
                status.textContent = container.getAttribute('data-label-
failed') || 'Sicherheitspruefung fehlgeschlagen. Bitte Seite neu laden.';
            }
        }
    }

    function guardForm(form) {
        form.addEventListener('submit', function (event) {
            var widgets = form.querySelectorAll('.fp-altcha');
            for (var i = 0; i < widgets.length; i++) {
                var state = widgets[i].getAttribute('data-fp-altcha-state');
                if (state !== 'verified') {
                    event.preventDefault();
                    var status = widgets[i].querySelector('.fp-altcha-
status');
                    if (status) {
                        status.textContent = 'Bitte einen Moment warten, die
Sicherheitspruefung laeuft noch.';
                    }
                    return;
                }
            }
        });
    }

    function init() {
        if (!window.crypto || !window.crypto.subtle) {
            // Sehr alte Browser ohne Web Crypto API: Pruefung kann nicht
durchgefuehrt werden.
            // Das Formular bleibt dann serverseitig blockiert (kein
gueltiges "altcha" Feld).
            return;
        }

        var containers = document.querySelectorAll('.fp-altcha[data-fp-
altcha]');
        containers.forEach(function (container) {
            processWidget(container);

            var form = container.closest('form');
```

```
            if (form && !form.hasAttribute('data-fp-altcha-guarded')) {
                form.setAttribute('data-fp-altcha-guarded', '1');
                guardForm(form);
            }
        });
    }

    if (document.readyState === 'loading') {
        document.addEventListener('DOMContentLoaded', init);
    } else {
        init();
    }
})();
```