

How to Store Some Secrets

Reto E. Koenig, Rolf Haenni

University of Fribourg
&
Bern University of Applied Sciences

12.07.2011

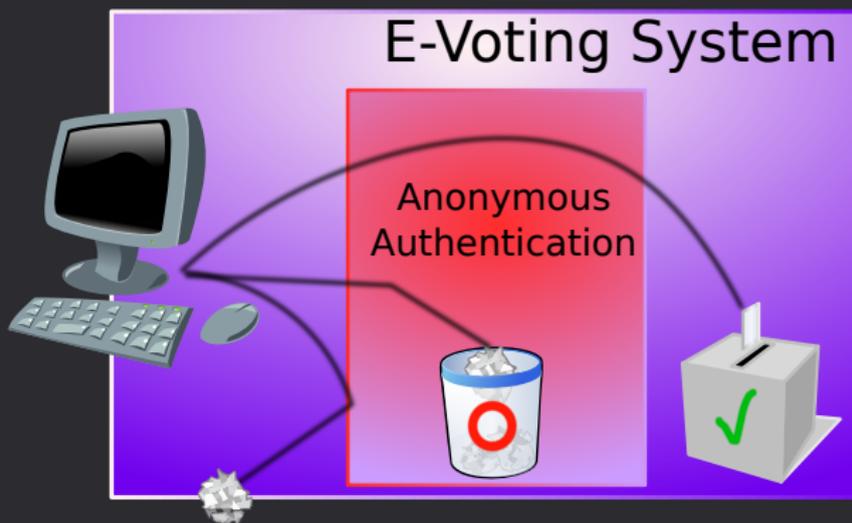


Usability Studies JCJ-05

The voter has to memorize different credentials with very high entropy:

Real The credential for the real voting act

Fake The credential used to deceive the adversary



Question

How to store and discriminate these credentials without hinting the adversary?

Hardening JCJ-05 for reality

Speedup JCJ-05 is too slow for large scale elections

Board flooding Easy to bring down JCJ-05 by a denial of service attack.

Mission accomplished, Problems solved.^[KHS11]

[KHS11] R. Koenig and R. Haenni and S. Fischli

Preventing Board Flooding Attacks in Coercion-Resistant Electronic Voting Schemes

SEC'11, 26th IFIP International Information Security

Hardening JCJ-05 for reality

Speedup JCJ-05 is too slow for large scale elections

Board flooding Easy to bring down JCJ-05 by a denial of service attack.

Mission accomplished, Problems solved.^[KHS11]

[KHS11] R. Koenig and R. Haenni and S. Fischli
Preventing Board Flooding Attacks in Coercion-Resistant Electronic Voting Schemes

SEC'11, 26th IFIP International Information Security

Usability Studies on the Hardened JCJ05 Derivate

Each voter...

- ...needs to secretly store several dozens credentials
- ...has to discriminate doubtless between credentials for 'Accept' and 'Fake'.
- ...is not allowed to mark any credential
- ...shall never unveil the amount of possessed secrets (They vary per voter)

```

924f61661a3472da74307a35f2c8d22e07e84a4d ○
cbf019b764b9477080c5a9a748a2911a5fa6d614 ○
fc8ccd6641d45ef2efdd926c3a6f7f3ac268e9e3 ✓
a29965fbb2954c8a66d856e8eb891bce5f49dacf ✓
3a710d2a84f856bc4e1c0bb93ca517893c48691 ○
e1a5b5d17e51d56f0d6fc060968ff238afb9b32 ○
cbf019b764b9477080caa9a748a2911a5fa6d614 ○
a29965fbb2954c8a66d856e8eb891bce5f49dacf ○
22eb60281c37e6611e85e7a432a45c8f3525749 ○
924f61661a34d2da74307a35f2c8d22e07e84a4d ○
f5abae583297649847c13be2c54bcfb3268f8f3 ○
blaa98ad3a02ffe896c49687300d8644f50fd088 ○
2789ae8b84e43e4d5b60d87b3d1edec02d4c449e ○
2789ae8b84e43e4d5b60d87d3d1edec02d4c449e ○
fc0ccd6641d45ef2efdd946c3a6f7f3ac268e9e3 ○
e1a5b5d17e51d56f0d6fc060e68ff238afb9b32 ○
fd8823d985947fc7d9f470907ca18ed68243557 ○
f5abae583297649547c13be2c54bcfb3268f8f3 ✓
3a710d2a84f856bcce1c0bb93ca517893c48691 ✓
blaa98ad3a02ffe896c49687300d8644f50fd088 ✓
22eb60281c37e6611e85e7a432a45c8f3525749 ○
fd8823d985947fc7d9f470907ca18ed68243557 ○

```

Usability Studies on the Hardened JCJ05 Derivate

Each voter...

- ...needs to secretly store several dozens credentials
- ...has to discriminate doubtless between credentials for 'Accept' and 'Fake'.
- ...is not allowed to mark any credential
- ...shall never unveil the amount of possessed secrets (They vary per voter)

```

924f61661a3472da74307a35f2c8d22e07e84a4d
cbf019b764b9477080c5a9a748a2911a5fa6d614
fc0ccd6641d45ef2efd926c3a6f7f3ac268e9e3
a29965fbb2954c8a66d856e8eb891bce5f49dacf
3a710d2a84f856bc4e1c0bb93ca517893c48691
e1a5b5d17e51d56f0d6fc060968ff238afb9b32
cbf019b764b9477080caa9a748a2911a5fa6d614
a29965fbb2954c8a66d856e8eb891bce5f49dacf
22eb60281c37e6611e85e7a432a45c8f3525749
924f61661a34d2da74307a35f2c8d22e07e84a4d
f5abae583297649847c13be2c54bcfb3268f0f3
b1aa98ad3a02ffe896c49687300d8644f50fd088
2789ae8b84e43e4d5b60d87b3d1edec02d4c449e
2789ae8b84e43e4d5b60d87d3d1edec02d4c449e
fc0ccd6641d45ef2efd946c3a6f7f3ac268e9e3
e1a5b5d17e51d56f0d6fc060e68ff238afb9b32
fd8b823d965947fc7d9f470907ca18ed68243557
f5abae583297649547c13be2c54bcfb3268f0f3
3a710d2a84f856bce1c0bb93ca517893c48691
b1aa98ad3a02ffe896c49687300d8644f50fd088
22eb60281c37e6611e85e7a432a45c8f3525749
fd8b823d965947fc7d9f470907ca18ed68243557

```

Usability Studies on the Hardened JCJ05 Derivate

Each voter...

- ...needs to secretly store several dozens credentials
- ...has to discriminate doubtless between credentials for 'Accept' and 'Fake'.
- ...is not allowed to mark any credential
- ...shall never unveil the amount of possessed secrets (They vary per voter)



Question



How to manage...

Strategies

- Password vault with a single master password
 - Challengeable 'offline'
 - Once open, every credential visible
- One ciphertext per credential
 - Managing ciphers
 - Match password and cipher... Which is what?
- Secret-Storage System
 - Well...

Strategies

- Password vault with a single master password
 - Challengeable 'offline'
 - Once open, every credential visible
- One ciphertext per credential
 - Managing ciphers
 - Match password and cipher... Which is what?
- Secret-Storage System
 - Well...

Strategies

- Password vault with a single master password
 - Challengeable 'offline'
 - Once open, every credential visible
- One ciphertext per credential
 - Managing ciphers
 - Match password and cipher... Which is what?
- Secret-Storage System
 - Well...

Strategies

- Password vault with a single master password
 - Challengeable 'offline'
 - Once open, every credential visible
- One ciphertext per credential
 - Managing ciphers
 - Match password and cipher... Which is what?
- Secret-Storage System
 - Well...

Properties of a Secret-Storage System

The system...

- ...allows to choose freely n keys
- ...allows to choose freely n secrets
- ...allows to store multiple secrets in **one** storage (aka cipher)
- ...allows to retrieve **only** the secret correlated to the key
- ...has all properties of a (symmetric) crypto-system



Definition of a Secret-Storage System

$$\Sigma[n] = (\mathcal{S}, \mathcal{K}, \mathcal{C}, \text{store}, \text{retrieve})$$

$\Sigma[n] = (\mathcal{S}, \mathcal{K}, \mathcal{C}, \text{store}, \text{retrieve})$

\mathcal{S} = *secret space*, set of all possible secrets

\mathcal{K} = *key space*, set of all possible keys

\mathcal{C} = *storage space*, the set of all possible storages

store : $\mathcal{S}^n \times \mathcal{K}^{(n)} \longrightarrow \mathcal{C}$

storage function, where $\mathcal{K}^{(n)} \subseteq \mathcal{K}^n$ is the set of all admissible key tuples (with distinct keys)

retrieve : $\mathcal{C} \times \mathcal{K} \longrightarrow \mathcal{S}$

the *retrieval function*

$\Sigma[n] = (\mathcal{S}, \mathcal{K}, \mathcal{C}, \text{store}, \text{retrieve})$

\mathcal{S} = *secret space*, set of all possible secrets

\mathcal{K} = *key space*, set of all possible keys

\mathcal{C} = *storage space*, the set of all possible storages

store : $\mathcal{S}^n \times \mathcal{K}^{(n)} \longrightarrow \mathcal{C}$

storage function, where $\mathcal{K}^{(n)} \subseteq \mathcal{K}^n$ is the set of all admissible key tuples (with distinct keys)

retrieve : $\mathcal{C} \times \mathcal{K} \longrightarrow \mathcal{S}$

the *retrieval function*

$S = (s_1, \dots, s_n) \in \mathcal{S}^n$, an n-tuple of secrets ($n \geq 1$)

$K = (k_1, \dots, k_n) \in \mathcal{K}^{(n)}$, an n-tuple of **distinct** keys $n \geq 1$

c = a particular storage

$\text{store}_K(S) = c \in \mathcal{C}$, storing the n-tuple of the secrets $S \in \mathcal{S}^n$
with the n-tuple of distinct keys $K \in \mathcal{K}^{(n)}$

$\text{retrieve}_{k_i}(c) = s_i \in \mathcal{S}$ retrieval with key k_i

$$\text{retrieve}_{k_i}(\text{store}_K(S)) = s_i$$

$S = (s_1, \dots, s_n) \in \mathcal{S}^n$, an n-tuple of secrets ($n \geq 1$)

$K = (k_1, \dots, k_n) \in \mathcal{K}^{(n)}$, an n-tuple of **distinct** keys $n \geq 1$

c = a particular storage

$\text{store}_K(S) = c \in \mathcal{C}$, storing the n-tuple of the secrets $S \in \mathcal{S}^n$
with the n-tuple of distinct keys $K \in \mathcal{K}^{(n)}$

$\text{retrieve}_{k_i}(c) = s_i \in \mathcal{S}$ retrieval with key k_i

$$\text{retrieve}_{k_i}(\text{store}_K(S)) = s_i$$

$S = (s_1, \dots, s_n) \in \mathcal{S}^n$, an n-tuple of secrets ($n \geq 1$)

$K = (k_1, \dots, k_n) \in \mathcal{K}^{(n)}$, an n-tuple of **distinct** keys $n \geq 1$

c = a particular storage

$\text{store}_K(S) = c \in \mathcal{C}$, storing the n-tuple of the secrets $S \in \mathcal{S}^n$
with the n-tuple of distinct keys $K \in \mathcal{K}^{(n)}$

$\text{retrieve}_{k_i}(c) = s_i \in \mathcal{S}$ retrieval with key k_i

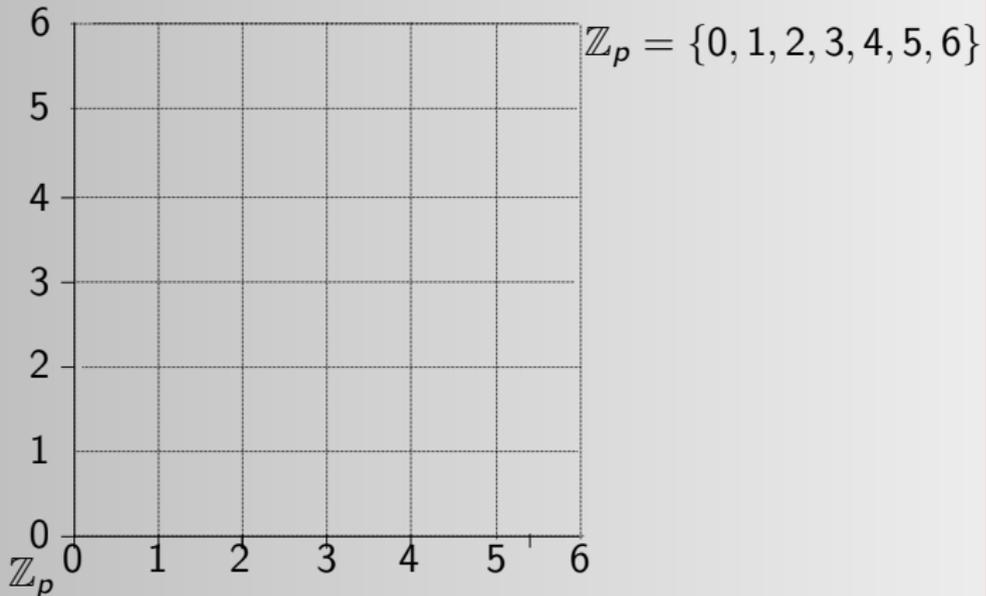
$$\text{retrieve}_{k_i}(\text{store}_K(S)) = s_i$$

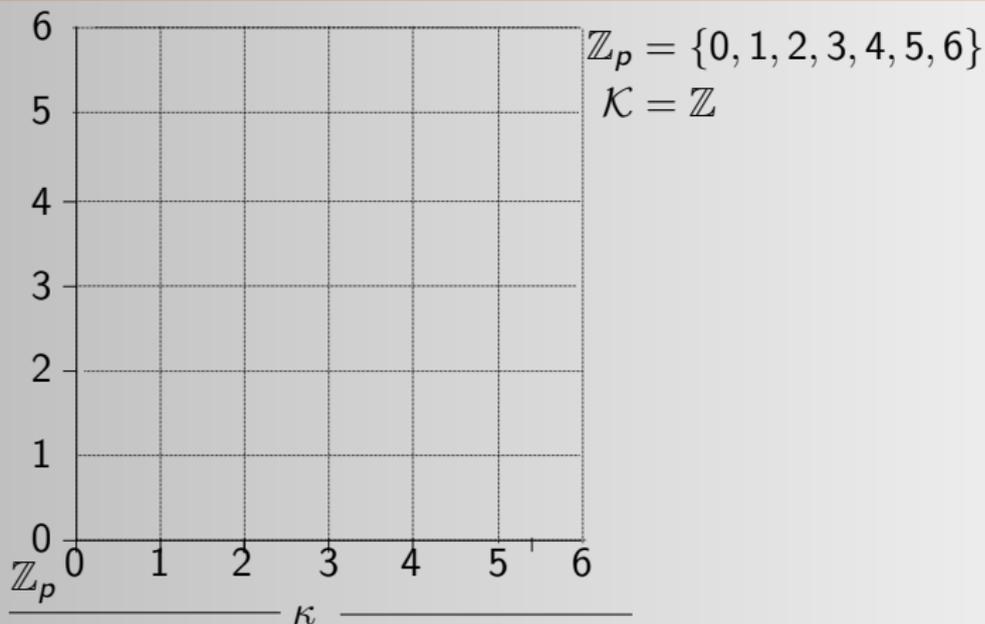
Properties of the Secret-Storing System

Required to possess the cryptographic properties of a conventional symmetric crypto-system:

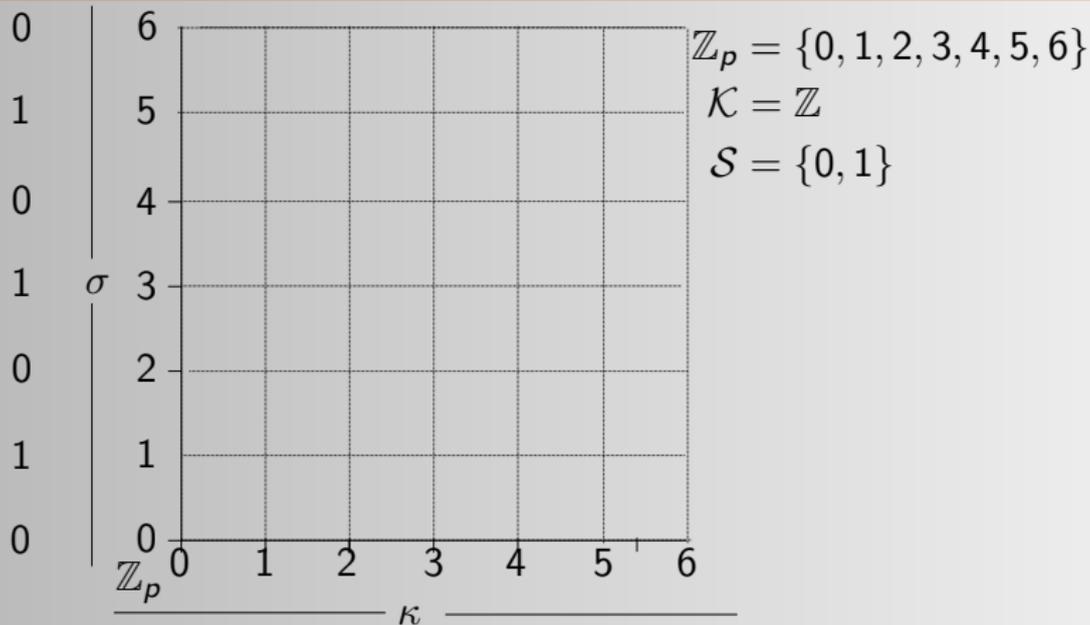
- Retrieving s_i from c does not disclose any information about the other secrets in c
- Applying K on c returns S
- Serves a conditional entropy $H(S|c)$ which is equal to $H(S)$
- Applying K' on c where $K' \neq K$ does return S with a probability of $\frac{1}{|S|}$

Realisation using a Prime Field \mathbb{Z}_p , where $p = 7, n = 3$

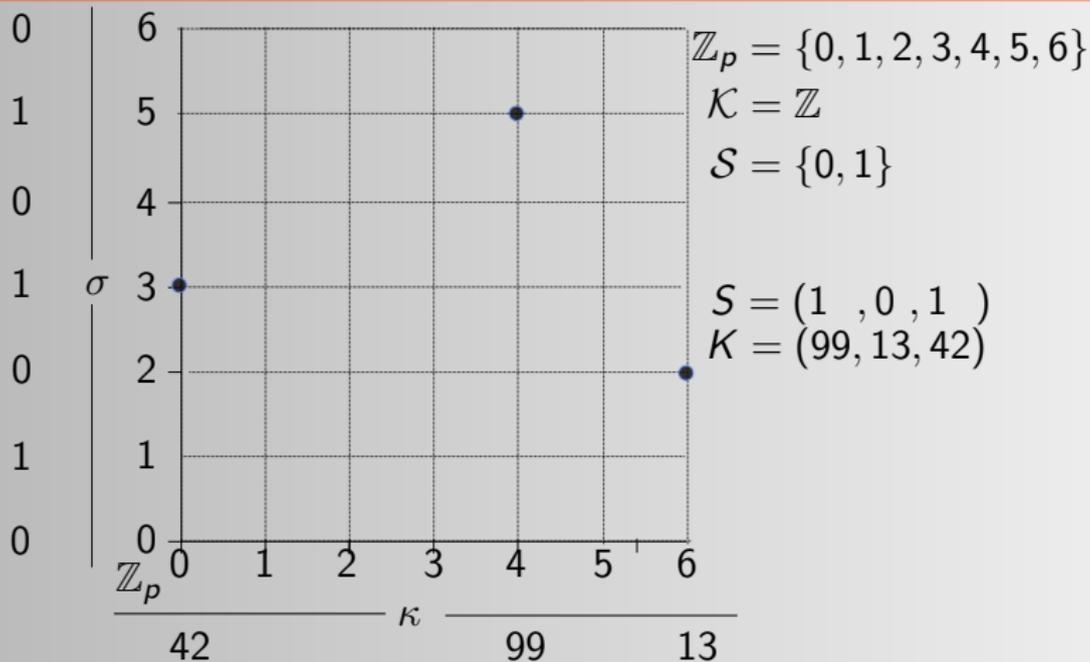


Realisation using a Prime Field \mathbb{Z}_p , where $p = 7, n = 3$ 

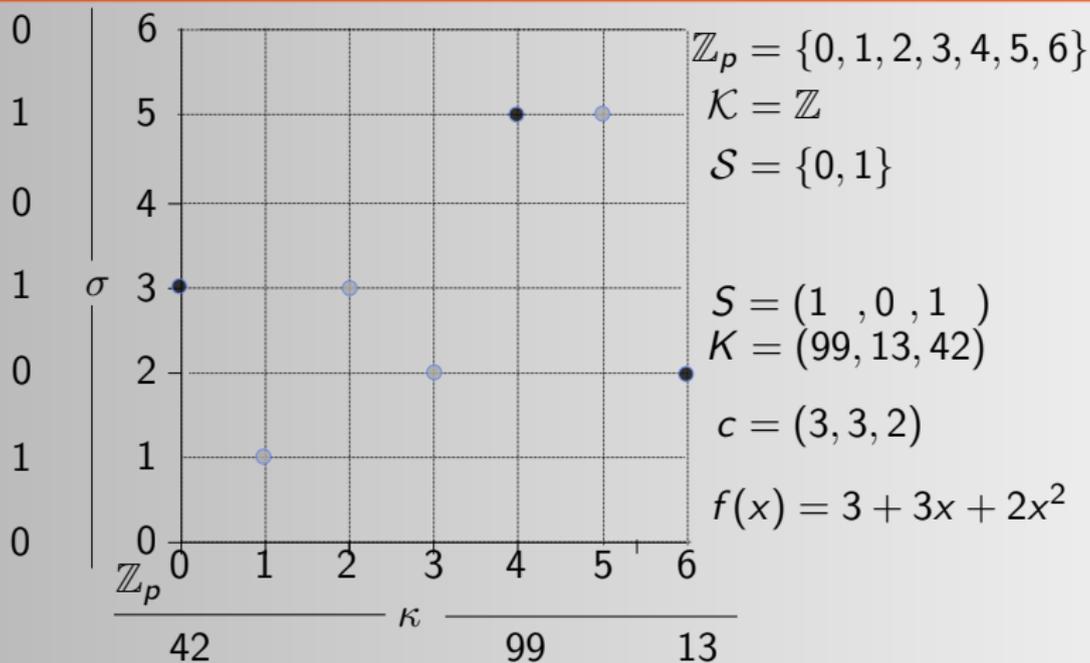
Realisation using a Prime Field \mathbb{Z}_p , where $p = 7, n = 3$



The *store*-Function in \mathbb{Z}_p , where $p = 7, n = 3$



The *store*-Function in \mathbb{Z}_p , where $p = 7, n = 3$

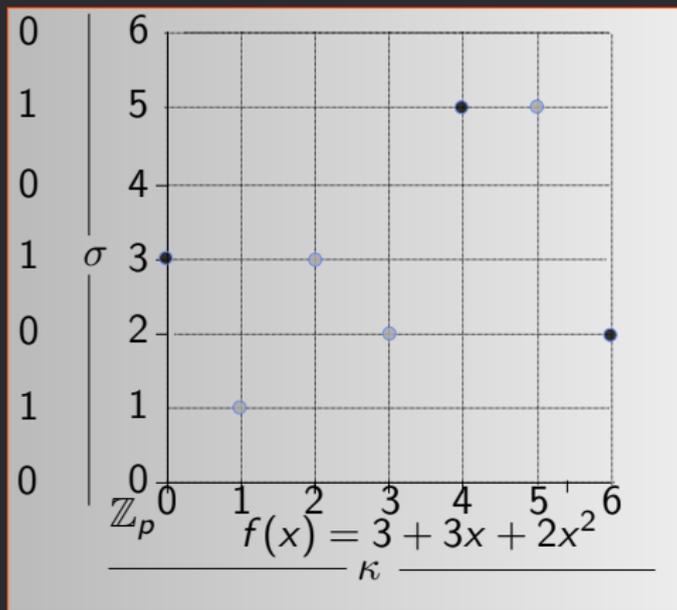


The *retrieve*-Function for the key 99 in \mathbb{Z}_p , where $p = 7, n = 3$

$$\mathbb{Z} \mapsto \mathbb{Z}_p \quad \kappa(99) = 4$$

$$f(x) \quad f(4) = 5$$

$$\mathbb{Z}_p \mapsto \mathcal{S} \quad \sigma(5) = 1$$



Now What?

