Clearing Fuzzy Signatures: a Proof of Work Blockchain Protocol for Biometric Identification

Paolo Santini, Giulia Rafaiani, Massimo Battaglioni, Franco Chiaraluce, Marco Baldi
Dipartimento di Ingegneria dell’Informazione
Università Politecnica delle Marche

5th Distributed Ledger Technology Workshop (DLT 2023)  Thursday 25th May 2023
Introduction

• Extracting secrets from a physical source is often tricky, since the source can be noisy.

• For authentication purposes, different readings of the same secret will be close, but still not identical, one to each other.

• When data from fuzzy sources are used as the secret key for multiple digital signatures, the resulting signatures will fail verification with the originally enrolled public key, if some techniques to reduce noise are not employed.
As for authentication, an increasing trend is that of relying on decentralization.

Existing fuzzy authentication schemes are not directly linked to the problem of reconciling with a stored template and use noise-reduction techniques, like error-correcting codes.
Aim of the work

- As for authentication, an increasing trend is that of relying on decentralization.
- Existing fuzzy authentication schemes are not directly linked to the problem of reconciling with a stored template and use noise-reduction techniques, like error-correcting codes.

Our aim is to create a decentralized fuzzy system for authentication purposes fully leveraging blockchain technology.
System features

- Users are simply required to *digitally sign* some random message using fuzzy keys
Users are simply required to *digitally sign* some random message using fuzzy keys.

The system will *not* use noise-reducing techniques.
System features

• Users are simply required to *digitally sign* some random message using fuzzy keys
• The system will *not* use noise-reducing techniques
• The blockchain will be *actively* part of the noise removal, providing the basis for a special instance of *Proof of Work* in which the mining process corresponds to the *de-noising process*
System features

- Users are simply required to *digitally sign* some random message using fuzzy keys.
- The system will *not* use noise-reducing techniques.
- The blockchain will be *actively* part of the noise removal, providing the basis for a special instance of *Proof of Work* in which the mining process corresponds to the de-noising process.
- We consider classic RSA digital signatures, showing that fuzziness in the secret key translates into some noise affecting the derived signatures.
Fuzzy Signature Scheme

- $sk$ is sampled from the discrete distribution $\mathcal{D}$ of the fuzzy source
Fuzzy Signature Scheme

- sk is sampled from the discrete distribution $\mathcal{D}$ of the fuzzy source
  - $\text{KeyGen}_\mathcal{D}()$: sample $sk \leftarrow \mathcal{D}$, then compute the corresponding public key $pk \in \mathcal{P}$;
Fuzzy Signature Scheme

- sk is sampled from the discrete distribution $\mathcal{D}$ of the fuzzy source
  - $\text{KeyGen}_\mathcal{D}()$: sample $sk \leftarrow \mathcal{D}$, then compute the corresponding public key $pk \in \mathcal{P}$;
  - $\text{Sign}_\mathcal{D}(m)$: on input a message $m \in \mathcal{M}$, sample $sk \leftarrow \mathcal{D}$, then run $\text{Sign}(m, sk)$ on input the sampled $sk$
Fuzzy Signature Scheme

- sk is sampled from the discrete distribution $\mathcal{D}$ of the fuzzy source
  - $\text{KeyGen}_\mathcal{D}():$ sample $\text{sk} \leftarrow \mathcal{D}$, then compute the corresponding public key $\text{pk} \in \mathcal{P}$;
  - $\text{Sign}_\mathcal{D}(m):$ on input a message $m \in \mathcal{M}$, sample $\text{sk} \leftarrow \mathcal{D}$, then run $\text{Sign}(m, \text{sk})$ on input the sampled $\text{sk}$

- The input secret key is another sample $\text{sk}'$ from the same fuzzy distribution. When $\text{sk}$ and $\text{sk}'$ are close, the associated signatures are also similar, according to some distance metric
Let \( \text{dist} : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}_+ \) be a distance function for which there exists some \( \theta \in \mathbb{R}_+ \) such that, for every pair of signatures \( \sigma, \sigma' \) on the same message \( m \), computed respectively with keys \( sk, sk' \) it holds that \( \text{dist}(\sigma, \sigma') \leq \theta \).
Let $\text{dist} : \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}_+$ be a distance function for which there exists some $\theta \in \mathbb{R}_+$ such that, for every pair of signatures $\sigma, \sigma'$ on the same message $m$, computed respectively with keys $sk, sk'$ it holds that $\text{dist}(\sigma, \sigma') \leq \theta$.

Then, we call ClearSignature an algorithm that, on input a triplet $(m, \sigma, pk)$, returns a signature $\sigma' \in \mathcal{S}$ such that $\text{dist}(\sigma, \sigma') \leq \theta$ and $\text{Verify}(m, \sigma', pk) = \text{True}$.
RSA Clear Signature

Let $p_i$ and $q_i$ be two primes, and $n_i = p_i q_i$. We define $\mathcal{D}_{S_i}$ as the distribution that returns samples of the form $x \equiv \hat{x}_i + e \mod n_i$, where $e$ is uniformly distributed over $[-w; w]$

\[
\text{RSA.ClearSign}(m, s, (n, \delta)):
\]
1) compute $c = \text{Hash}(m)$;
2) compute $y \equiv s^\delta \mod n$;
3) sample $\Delta e \leftarrow [−2w; 2w]$;
4) compute $\hat{c} \equiv ye^{-\delta \Delta e} \mod n$;
5) return $\Delta e$ if $\hat{c} = c$, else restart from Step 3.
System procedure

Distributed Network

Source $S_i$

Enter string $S_i$

Recover $n_i$ from $L$

Set $m = (S_i || \text{Context})$
Compute $s = \text{Sign} \mathcal{D}_i(m)$

RawTx = $(m, s)$

Run RSA.ClearSign$(m, s, pk_i)$
If $\Delta e \in [-2w; 2w]$ is found:
Set MinedTx = $(m, s, \Delta e)$
Modified RSA Clear Signature

• If users collude, malicious miners can skip the clearing process (since they know the secret keys and, so, $\Delta e$) and produce blocks faster than honest miners, which execute RSA.ClearSign
If users collude, malicious miners can skip the clearing process (since they know the secret keys and, so, $\Delta e$) and produce blocks faster than honest miners, which execute RSA.ClearSign

\[
\text{RSA.ClearSign}^{(\text{PRNG})}(m, \text{aux}, s, (n, \delta)):
\begin{align*}
1) & \text{ compute } c = \text{Hash}(m); \\
2) & \text{ compute } y \equiv s^\delta \mod n; \\
3) & \text{ sample } \text{seed} \leftarrow \mathcal{R}; \\
4) & \text{ compute } \Delta e = \text{PRNG}(m||\text{seed}||\text{aux}); \\
5) & \text{ compute } \hat{c} \equiv yc^{-\Delta e} \mod n; \\
6) & \text{ return } \text{seed} \text{ if } \hat{c} = c, \text{ else restart from Step 3.}
\end{align*}
\]
Byzantine Fault Tolerance

- If all malicious miners $\tilde{M}$ know in advance the value of $\Delta e$, but do not know its pre-image seed, all works until

$$\frac{(4w + 1)t_{PRNG}}{\tilde{M}} > \frac{(4w + 1)(t_{PRNG} + t_{RSA})}{M - \tilde{M}}$$
Byzantine Fault Tolerance

- If all malicious miners $\tilde{M}$ know in advance the value of $\Delta e$, but do not know its pre-image seed, all works until

$$\frac{(4w + 1)t_{PRNG}}{\tilde{M}} > \frac{(4w + 1)(t_{PRNG} + t_{RSA})}{M - \tilde{M}}$$

$$\tilde{M} < M \cdot \frac{t_{PRNG}}{2t_{PRNG} + t_{RSA}} = M \cdot \frac{1}{2 + \frac{t_{RSA}}{t_{PRNG}}}$$
Byzantine Fault Tolerance

- If all malicious miners $\tilde{M}$ know in advance the value of $\Delta e$, but do not know its pre-image seed, all works until

$$\frac{(4w + 1)t_{\text{PRNG}}}{\tilde{M}} > \frac{(4w + 1)(t_{\text{PRNG}} + t_{\text{RSA}})}{M - \tilde{M}}$$

$$\tilde{M} < M \cdot \frac{t_{\text{PRNG}}}{2t_{\text{PRNG}} + t_{\text{RSA}}} = M \cdot \frac{1}{2 + \frac{t_{\text{RSA}}}{t_{\text{PRNG}}}}$$

- **PRNG cannot be much more efficient than RSA**
Conclusion

• The authentication process is delegated to a distributed network and executes the task of removing noise from fuzzy signatures
Conclusion

• The authentication process is delegated to a distributed network and executes the task of removing noise from fuzzy signatures

• The design of an ad-hoc PRNG leads to $BFT \sim \frac{\tilde{M}}{M} \approx \frac{1}{2^{1/X}}$
Conclusion

• The authentication process is delegated to a distributed network and executes the task of removing noise from fuzzy signatures.

• The design of an ad-hoc PRNG leads to $BFT \sim \frac{\bar{M}}{M} \approx \frac{1}{2^{1/X}}$.

• Mining and verification times grow respectively as $O\left(\frac{X+1}{M}\right)$ and $O(X)$. 

Conclusion

• The authentication process is delegated to a distributed network and executes the task of removing noise from fuzzy signatures

• The design of an ad-hoc PRNG leads to $\text{BFT} \sim \frac{\bar{M}}{M} \approx \frac{1}{2^{1/X}}$

• Mining and verification times grow respectively as $O\left(\frac{X+1}{M}\right)$ and $O(X)$

• Security analysis and application to RSA signature scheme show the feasibility of the approach
Thanks for your kind attention!

g.rafaiani@univpm.it