



Technische
Universität
Braunschweig



Nutzung von Kontextinformationen zur Herstellung eines sicheren Kommunikationskanals

Bachelorarbeit

Dominik Schürmann

16. November 2010

Inhalt

Einführung

Audio-Fingerprinting

Fuzzy Cryptography

Fuzzy Pairing

Zusammenfassung

Idee

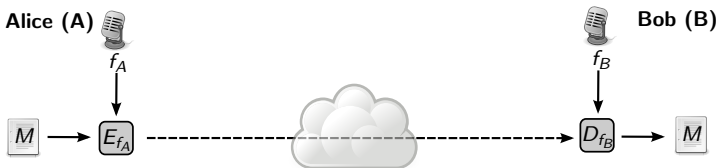
Problemstellung

- Zwei drahtlose Endgeräte sollen ad hoc eine verschlüsselte Verbindung herstellen
- Kein manueller Schlüsselaustausch
- Generierung der Schlüssel durch Kontextinformationen

Konkreter

- Kontextinformationen $\hat{=}$ zeitveränderliche Audiodaten (Umgebungsgeräusche)

Umsetzungsidee

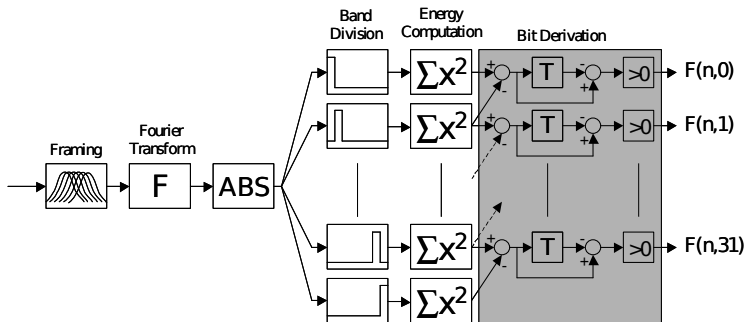


f_A, f_B Audio-Fingerprints der lokalen Umgebungsgeräusche von Alice A und Bob B

E_{f_A}, D_{f_B} Chiffrierungs- und Dechiffrierungsfunktion unter Nutzung der Fingerprints

M Zu sendende Nachricht

Audio-Fingerprinting nach Haitsma und Kalker¹



- Fingerprint-Algorithmus basierend auf Energie-Differenzen
- Fingerprint wird durch Bitsequenz repräsentiert

¹Jaap Haitsma und Ton Kalker. „A Highly Robust Audio Fingerprinting System“. In: *Journal of New Music Research* 32.2 (2003), S. 211–221.

Vereinfachte Implementierung

Aufteilung

- ~ 3 s Audiosequenz, aufgeteilt in $8 \cdot \sim 0,37$ s Frames
- Pro Frame 33 Frequenzbänder

Berechnung des Audio-Fingerprints

$E(n, m)$ Energie des Frames n auf dem Frequenzband m

$F(n, m)$ Fingerprint des m -ten Frequenzbands des Frames n

Für $n = 1, \dots, 8$:

Für $m = 1, \dots, 32$:

$$F(n, m) = \begin{cases} 1 & \text{falls } E(n, m) - E(n, m+1) - (E(n-1, m) - E(n-1, m+1)) > 0, \\ 0 & \text{falls } E(n, m) - E(n, m+1) - (E(n-1, m) - E(n-1, m+1)) \leq 0. \end{cases}$$

\Rightarrow 32 Bit Subfingerprint pro Frame \Rightarrow 256 Bit Fingerprint für ~ 3 s

Probleme im Realeinsatz

Durchschnittliche Übereinstimmung der Fingerprints von unterschiedlichen Audiosequenzen

		Mikrofon B				
		clap1	music1	snap1	speak1	whistle1
Mikrofon A	clap1	87,5	53,91	55,08	51,56	52,54
	music1	50,59	76,37	49,8	53,32	50,0
	snap1	55,27	50,98	78,71	50,2	51,56
	speak1	51,17	53,91	49,22	80,08	56,84
	whistle1	54,1	50,98	53,71	53,71	77,73

		Mikrofon B				
		speak1	speak2	speak3	speak4	speak5
Mikrofon A	speak1	80,08	51,17	53,52	51,76	55,86
	speak2	50,0	76,56	52,34	54,1	57,81
	speak3	51,95	54,69	71,48	53,71	50,39
	speak4	51,76	50,2	57,23	79,3	55,27
	speak5	54,49	54,49	53,71	55,08	79,88

- Gute Ergebnisse. . .
- . . . aber
Übereinstimmung
nicht hoch genug als
dass die Fingerprints
als eindeutige
Schlüssel genutzt
werden könnten

Tabellenwerte: Angaben in Prozent

Abstände zur Quelle Q : $\Delta(A, Q) = 1,5m$ und $\Delta(B, Q) = 3m$

Fuzzy Cryptography

Gesucht

- Kryptografisches Verfahren, das gewisse Differenzen toleriert
- Threshold soll flexibel gewählt werden können
- Direkte Nutzung der Audio-Fingerprints

Fuzzy Cryptography

Gesucht

- Kryptografisches Verfahren, das gewisse Differenzen toleriert
- Threshold soll flexibel gewählt werden können
- Direkte Nutzung der Audio-Fingerprints

Lösung

- Fuzzy Cryptography basierend auf Fehlerkorrigierenden Codes

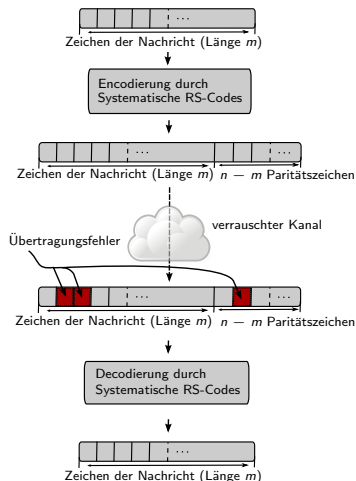
Reed-Solomon-Codes

Encodierung

Eine Nachricht $a \in \mathcal{A} = \mathbb{F}_q^m$ wird mit einem Reed-Solomon-Code $RS(q, m, n)$ auf ein Codewort $c \in \mathcal{C} = \mathbb{F}_q^n$ abgebildet, wobei $q = 2^p$ und p prim.

Decodierung bei Fehlern

Das fehlerhafte Codewort $c' \in \mathcal{C}$ wird wieder zu dem ursprünglich gesendeten $a \in \mathcal{A}$ decodiert, solange $\leq \lfloor \frac{n-m}{2} \rfloor$ Übertragungsfehler aufgetreten sind.



Beispiel Reed-Solomon-Codes

Reed-Solomon-Code: $RS(2^3, 3, 5) = RS(8, 3, 5)$

$\Rightarrow \mathcal{A} = \mathbb{F}_8^3$ und $\mathcal{C} = \mathbb{F}_8^5$

\Rightarrow Correction Threshold $t = \lfloor \frac{n-m}{2} \rfloor = \lfloor \frac{5-3}{2} \rfloor = 1$

Beispiel Reed-Solomon-Codes

Reed-Solomon-Code: $RS(2^3, 3, 5) = RS(8, 3, 5)$

$\Rightarrow \mathcal{A} = \mathbb{F}_8^3$ und $\mathcal{C} = \mathbb{F}_8^5$

\Rightarrow Correction Threshold $t = \lfloor \frac{n-m}{2} \rfloor = \lfloor \frac{5-3}{2} \rfloor = 1$

Encodierung

- $a = 0, 6, 5$
- $c = \text{Encode}(a) = 0, 6, 5, 5, 2$

Beispiel Reed-Solomon-Codes

Reed-Solomon-Code: $RS(2^3, 3, 5) = RS(8, 3, 5)$

$\Rightarrow \mathcal{A} = \mathbb{F}_8^3$ und $\mathcal{C} = \mathbb{F}_8^5$

\Rightarrow Correction Threshold $t = \lfloor \frac{n-m}{2} \rfloor = \lfloor \frac{5-3}{2} \rfloor = 1$

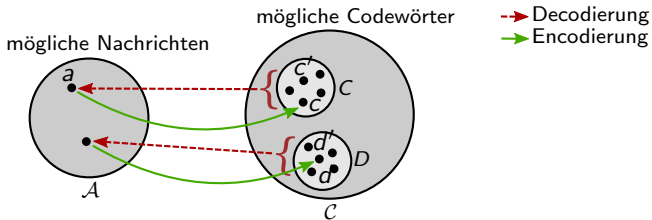
Encodierung

- $a = 0, 6, 5$
- $c = \text{Encode}(a) = 0, 6, 5, 5, 2$

Decodierung

- $c' = 0, 6, 5, \mathbf{3}, 2$
- $a = \text{Decode}(c') = 0, 6, 5$

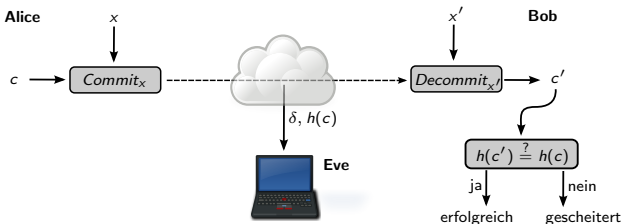
Schematische Darstellung der Funktionsweise



Encodierung Eindeutige Abbildung $a \mapsto c : a \in \mathcal{A}, c \in \mathcal{C}$

Decodierung $\forall \tilde{c} \in \mathcal{C} : \tilde{c} \mapsto a : a \in \mathcal{A}, \mathcal{C} = \{c, c', c'', \dots\} \in \mathcal{C}$

Fuzzy Commitment basierend auf Juels und Wattenberg²



Wörter Codewörter $c, c' \in \mathcal{C}$, Geheimnis $x, x' \in \mathcal{C}$

h Geeignete Hashfunktion, z. B. SHA-256

Commit_x Berechnung $\delta = x - c$, wobei jeder einzelne Wert durch $\cdot \bmod q$ wieder in \mathcal{C} abgebildet wird.

Decommitt_{x'} Berechnung $\tilde{c} = x' - \delta$. Decodierung bildet $\tilde{c} \in \mathcal{C}$ auf ein $a \in \mathcal{A}$ ab. Die Encodierung wiederum bildet dieses a eindeutig auf ein $c' \in \mathcal{C}$ zurück.

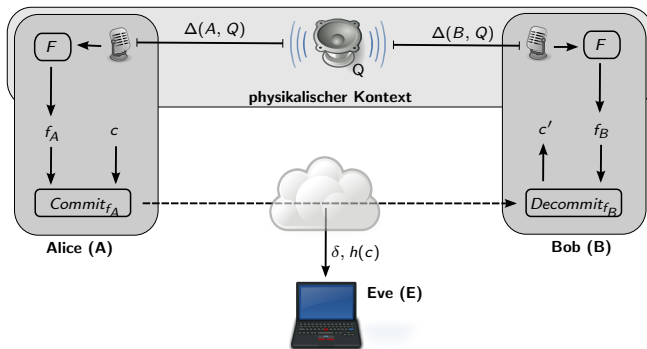
²Ari Juels und Martin Wattenberg. „A Fuzzy Commitment Scheme“. In: *Sixth ACM Conference on Computer and Communications Security* (1999), S. 28–36.

Eigene Entwicklungen

- Umsetzung des Fuzzy Commitments mit Reed-Solomon-Codes als Schlüsselverteilungsverfahren
- Thresholds für unterschiedlich starke Anforderungen an den Kontext
- Nutzung der Fingerprints f_A, f_B als Geheimnisse $x, x' \in \mathcal{C}$
- Versuche im realen Einsatz

⇒ Entwicklung eines Pairing-Protokolls

Modell des Fuzzy Pairings



- F** Audio-Fingerprinting-Methode nach Haitsma und Kalker, angepasst auf die Problemstellung
- c** Ausgangswort $a \in_R \mathcal{A}$ wird zufällig gewählt und durch einen Reed-Solomon-Code eindeutig zu $c \in \mathcal{C}$ encodiert

Beispiel Fuzzy Pairing

Generierung der Fingerprints

- Fingerprint von Alice: $f_A = 1, 0, 1, 0, 0$
- Fingerprint von Bob: $f_B = 1, 0, 1, \mathbf{1}, 0$

Beispiel Fuzzy Pairing

Generierung der Fingerprints

- Fingerprint von Alice: $f_A = 1, 0, 1, 0, 0$
- Fingerprint von Bob: $f_B = 1, 0, 1, \mathbf{1}, 0$

Initialisierung des Fuzzy Pairings

- Genutzter Reed-Solomon-Code: $RS(2^3, 3, 5) = RS(8, 3, 5)$
 - $\Rightarrow \mathcal{A} = \mathbb{F}_8^3$ und $\mathcal{C} = \mathbb{F}_8^5$
 - \Rightarrow Correction Threshold $t = \lfloor \frac{n-m}{2} \rfloor = \lfloor \frac{5-3}{2} \rfloor = 1$

Beispiel Fuzzy Pairing

Generierung der Fingerprints

- Fingerprint von Alice: $f_A = 1, 0, 1, 0, 0$
- Fingerprint von Bob: $f_B = 1, 0, 1, \mathbf{1}, 0$

Initialisierung des Fuzzy Pairings

- Genutzter Reed-Solomon-Code: $RS(2^3, 3, 5) = RS(8, 3, 5)$
 $\Rightarrow \mathcal{A} = \mathbb{F}_8^3$ und $\mathcal{C} = \mathbb{F}_8^5$
 \Rightarrow Correction Threshold $t = \lfloor \frac{n-m}{2} \rfloor = \lfloor \frac{5-3}{2} \rfloor = 1$

Start durch Alice

- Wähle zufälliges $a \in_R \mathcal{A}$ und encodiere dieses zu $c = \text{Encode}(a)$
 $\Rightarrow a = 0, 6, 5 \Rightarrow c = \text{Encode}(a) = 0, 6, 5, 5, 2$

Beispiel Fuzzy Pairing

Alice: Commit_{f_A}

- Berechnung des δ :

$$\begin{array}{r}
 f_A \quad 1, 0, 1, 0, 0 \\
 -_{(8)} \quad c \quad 0, 6, 5, 5, 2 \quad (\text{Sei } -_{(n)} \text{ die Subtraktion modulo } n) \\
 \hline
 \delta \quad 1, 2, 4, 3, 6
 \end{array}$$

Beispiel Fuzzy Pairing

Alice: Commit_{f_A}

- Berechnung des δ :

$$\begin{array}{r}
 f_A \quad 1, 0, 1, 0, 0 \\
 \text{---}_{(8)} \quad c \quad 0, 6, 5, 5, 2 \quad (\text{Sei } \text{---}_{(n)} \text{ die Subtraktion modulo } n) \\
 \delta \quad 1, 2, 4, 3, 6
 \end{array}$$

- Berechnung des Hashwerts $h(c)$ durch SHA-256:

$h(c) = '654ca94ae1baf4336e525b062f57d10a295bd7acf824102be4ec17999ac63162'$

Beispiel Fuzzy Pairing

Alice: Commit_{f_A}

- Berechnung des δ :

$$\begin{array}{r}
 f_A \quad 1, 0, 1, 0, 0 \\
 \text{---}_{(8)} \quad c \quad 0, 6, 5, 5, 2 \quad (\text{Sei } \text{---}_{(n)} \text{ die Subtraktion modulo } n) \\
 \hline
 \delta \quad 1, 2, 4, 3, 6
 \end{array}$$

- Berechnung des Hashwerts $h(c)$ durch SHA-256:

$$h(c) = '654ca94ae1baf4336e525b062f57d10a295bd7acf824102be4ec17999ac63162'$$

$\Rightarrow \delta, h(c)$ wird an Bob gesendet

Beispiel Fuzzy Pairing

Bob: $Decommit_{f_B}$

- Berechnung:

$$\begin{array}{r}
 f_B \quad 1, 0, 1, \mathbf{1}, 0 \\
 -_{(8)} \quad \delta \quad 1, 2, 4, 3, 6 \quad (\text{Sei } -_{(n)} \text{ die Subtraktion modulo } n) \\
 \hline
 \tilde{c} \quad 0, 6, 5, \mathbf{6}, 2
 \end{array}$$

Beispiel Fuzzy Pairing

Bob: $Decommit_{f_B}$

- Berechnung:

$$\begin{array}{r}
 f_B \quad 1, 0, 1, \mathbf{1}, 0 \\
 -_{(8)} \quad \delta \quad 1, 2, 4, 3, 6 \quad (\text{Sei } -_{(n)} \text{ die Subtraktion modulo } n) \\
 \hline
 \tilde{c} \quad 0, 6, 5, \mathbf{6}, 2
 \end{array}$$

- Decodierung durch RS-Codes: $a' = Decode(\tilde{c}) = 0, 6, 5$
- Encodierung durch RS-Codes: $c' = Encode(a') = 0, 6, 5, 5, 2$

Beispiel Fuzzy Pairing

Bob: $Decommit_{f_B}$

- Berechnung:

$$\begin{array}{r}
 f_B \quad 1, 0, 1, \mathbf{1}, 0 \\
 -_{(8)} \quad \delta \quad 1, 2, 4, 3, 6 \quad (\text{Sei } -_{(n)} \text{ die Subtraktion modulo } n) \\
 \hline
 \tilde{c} \quad 0, 6, 5, \mathbf{6}, 2
 \end{array}$$

- Decodierung durch RS-Codes: $a' = Decode(\tilde{c}) = 0, 6, 5$
- Encodierung durch RS-Codes: $c' = Encode(a') = 0, 6, 5, 5, 2$
- Berechnung des Hashwerts $h(c')$ durch SHA-256:
 $h(c') = '654ca94ae1baf4336e525b062f57d10a295bd7acf824102be4ec17999ac63162'$

Beispiel Fuzzy Pairing

Bob: $Decommit_{f_B}$

- Berechnung:

$$\begin{array}{r}
 f_B \quad 1, 0, 1, \mathbf{1}, 0 \\
 -_{(8)} \quad \delta \quad 1, 2, 4, 3, 6 \quad (\text{Sei } -_{(n)} \text{ die Subtraktion modulo } n) \\
 \hline
 \tilde{c} \quad 0, 6, 5, \mathbf{6}, 2
 \end{array}$$

- Decodierung durch RS-Codes: $a' = Decode(\tilde{c}) = 0, 6, 5$
- Encodierung durch RS-Codes: $c' = Encode(a') = 0, 6, 5, 5, 2$
- Berechnung des Hashwerts $h(c')$ durch SHA-256:
 $h(c') = '654ca94ae1baf4336e525b062f57d10a295bd7acf824102be4ec17999ac63162'$
- $h(c) \stackrel{?}{=} h(c') \Rightarrow$ Erfolgreiches Decommitment

Tests unter realen Bedingungen

Experimentelle Untersuchung

- Wahl der Fingerprintgrößen
- Wahl der passenden Parameter für die Initialisierung der Reed-Solomon-Codes, basierend auf Threshold und Fingerprintgröße

Ergebnisse

- 512 Bit Fingerprints durch ~ 6 s Audiosequenz
- Threshold für minimale Übereinstimmung der Fingerprints von 65%
 $\Rightarrow m = 180$

$\Rightarrow RS(2^{10}, 180, 512)$

Einsatz

Key Agreement

- Nutzung der Codewörter c, c' als Schlüssel für eine herkömmliche Chiffrierung, wie beispielsweise AES
- Verschlüsselte Ad-hoc-Kommunikation zwischen mobilen Endgeräten, Sensorknoten, ...

Ausblick

- Untersuchung der Entropie der Audio-Fingerprints
- Nutzung des Fuzzy Pairings mit anderen Kontextinformationen

Fragen?

Dominik Schürmann

`d.schuermann@tu-braunschweig.de`

Durchschnittliche prozentuale Übereinstimmungen

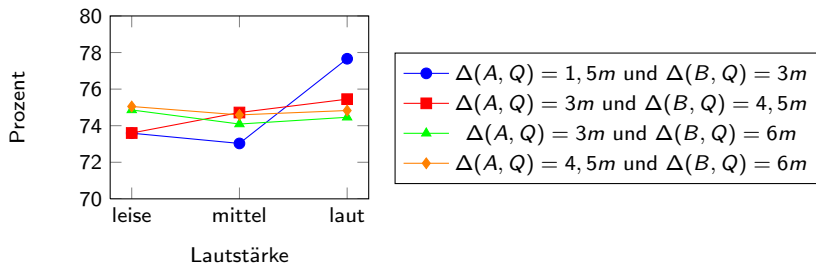


Abbildung: Durchschnittliche prozentuale Übereinstimmungen der zusammengehörigen Fingerprints in Abhängigkeit der Lautstärke

Correction Thresholds

m	t	minimale Übereinstimmung der Fingerprints
100	$\lfloor \frac{512-100}{2} \rfloor = 206$	$1 - \frac{206}{512} \approx 59,77\% \approx 60\%$
128	$\lfloor \frac{512-128}{2} \rfloor = 192$	$1 - \frac{192}{512} = 62,5\%$
152	$\lfloor \frac{512-152}{2} \rfloor = 180$	$1 - \frac{180}{512} \approx 64,84\% \approx 65\%$
204	$\lfloor \frac{512-204}{2} \rfloor = 154$	$1 - \frac{154}{512} \approx 69,92\% \approx 70\%$

Tabelle: Sinnvolle Werte m für eine Initialisierung durch $RS(2^{10}, m, 512)$, die dazugehörigen Correction Thresholds t und eine daraus resultierende minimale prozentuale Übereinstimmung der Fingerprints für ein erfolgreiches Pairing.

Synchronisationsprobleme

NTP

- Nutzung von NTP als Zeitsynchronisationsdienst ermöglicht Genauigkeit der Zeit mit einer Maximalabweichung von $\pm 0,01$ s (nach RFC5905)

Audio-Fingerprints mit unterschiedliche Startzeiten

- Neben dem Fingerprint $f_B = f_0$ mit Startzeit τ_{Start} : Weitere 100 Fingerprints f_i , die jeweils $\tau_{Start} + i * 0,001$ s als Startzeit besitzen, und 100 Fingerprints f_{-j} , wobei die Startzeit $\tau_{Start} + j * -0,001$ s beträgt

⇒ Ausgleich von maximal $\pm 0,1$ s Zeitdifferenz in der Startzeit τ_{Start} zwischen Alice und Bob