Token generation & security

supervised by Marius Lombard-Platet

Agathe Blaise\textsuperscript{1}, Paul Huynh\textsuperscript{2}, Diane Leblanc-Albarel\textsuperscript{3}, Simon Masson\textsuperscript{4}, Cyrius Nugier\textsuperscript{5}, Brice Wandji\textsuperscript{6}

\textsuperscript{1} Thales & Sorbonne Université - Paris, France
\textsuperscript{2} Université de Lorraine, INRIA, Loria, CNRS - Nancy, France
\textsuperscript{3} CNRS, Irisa - Rennes, France
\textsuperscript{4} Thales & Université de Lorraine, INRIA, Loria, CNRS - Nancy, France
\textsuperscript{5} Université de Toulouse, LAAS - CNRS, Toulouse, France
\textsuperscript{6} Fime - ENSICAEN, GREYC, Caen, France
441 billion purchase transactions on payment cards worldwide in 2019
441 billion purchase transactions on payment cards worldwide in 2019

Secure storing not possible for online payment: risk of data leak

→ Need for an alternative format to avoid storing credit card numbers
Card number tokenization

• Credit card number → string of random digits

• (De)tokenization = Token Service Provider (TSP)
Use case

1. Token request
2. Data retrieval
3. Token emission
4. Payment attempt
5. Payment request
6. Detokenization
7. Payment

Customer → TSP → Card issuer

1. #TOK
2. #CB
3. #TOK
4. #TOK
5. #TOK
6. #CB
7. Payment

Time-sensitive oper paste
Specifications

- Unicity
- Expiry time
- Format
- Identical distribution
- Unlinkability
- Timeframe
- Unforgeability
Specifications

• Unicity
• Expiry time
• Format
• Identical distribution
• Unlinkability
• Timeframe
• Unforgeability

1. Max. # of uses
2. Expiry time
Specifications

- Unicity
- Expiry time

Format

- Identical distribution
- Unlinkability
- Timeframe
- Unforgeability
Specifications

- Unicity
- Expiry time
- Format

- Identical distribution

- Unlinkability
- Timeframe
- Unforgeability
Specifications

- Unicity
- Expiry time
- Format
- Identical distribution

- Unlinkability

- Timeframe
- Unforgeability
Specifications

- Unicity
- Expiry time
- Format
- Identical distribution
- Unlinkability

- **Timeframe (100ms)**

- Unforgeability
Specifications

- Unicity
- Expiry time
- Format
- Identical distribution
- Unlinkability
- Timeframe

- Unforgeability
Primary Account Number
[ISO/IEC 7812]

- Issuer Identification Number (8)
- Individual Account Number (1 to 10, usually 7)
- Check Digit (1)
Token format

- TSP Number (4)
- Individual Token Number (8)
- Fixed digits + Checksum (4)

16 38 7
Main goals

- Quick detection of collisions during token generation
- Uniformity of the generated tokens
- Untraceability of the different tokens generated for the same user
Main goals

• Quick detection of **collisions** during token generation  

• **Uniformity** of the generated tokens  

• **Untraceability** of the different tokens generated for the same user  
Outline

• Toolbox
• Inconclusive approaches
• Our Free-spot-tokenization approach
• Extra possibilities
• Demo
• Conclusion
Toolbox

Hash function

- Collision-resistant
- Uniform output
Toolbox
HMAC function

- Collision-resistant
- Uniform output
Toolbox

Advanced Encryption Standard-256 (AES-256)
Inconclusive approaches

• Format-Preserving Encryption (FPE)
• Blockchain
Inconclusive approaches
Format-Preserving Encryption

Values with correct format

Values with correct format
Inconclusive approaches
Format-Preserving Encryption
Inconclusive approaches
Format-Preserving Encryption

Key (256 bits)

Domain

Domain
Inconclusive approaches
Format-Preserving Encryption

Your domain should be this big to be secure

Our Domain
Inconclusive approaches
Format-Preserving Encryption

Your domain should be this big to be secure

Our Domain

Soon?

2019
Inconclusive approaches
Format-Preserving Encryption

- $10^8$ tokens
  - same key: traceable
  - different key: collisions
- $< 10^8$ tokens: collisions
Inconclusive approaches
Blockchain in Tokenization Systems

- Distributed systems
- Trust
- Data integrity
- Transactions
Inconclusive approaches
Blockchain in Tokenization Systems

![Diagram showing blockchain transactions with labels: Previous Hash, Create Token 1234, Remove Token 5678]
Our Free-Spot-Tokenization approach
Tokenization

<table>
<thead>
<tr>
<th>ID</th>
<th>name</th>
<th>card number</th>
<th>conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Tokenization

data
Tokenization

data  HMAC

mod $10^8$

existing data
Tokenization

data + rand

HMAC

01001101
00110111
...
01101101

224 bits

mod 10^8

49203267

8 digits

@49203267

data

existing data
Tokenization

data’ + rand

HMAC

10011001
00001101
...
01001100

mod 10^8

57820183

224 bits

8 digits

data

Existing data

@57820183
Tokenization

data' + rand → HMAC → 10011001 00001101 ...
01001100

mod $10^8$ → 57820183

8 digits

@57820183

existing data

Existing data

57820183
Detokenization

@49203267
# 49203267

@ 47291271
# 47291271

@ 57820183
# 57820183

data

expired token

null
## Clean table

<table>
<thead>
<tr>
<th>@49203267</th>
<th>data</th>
</tr>
</thead>
<tbody>
<tr>
<td>@47291271</td>
<td>expired token</td>
</tr>
<tr>
<td>@57820183</td>
<td>null</td>
</tr>
</tbody>
</table>

## Clean-up
## Clean table

<table>
<thead>
<tr>
<th>@49203267</th>
<th>data</th>
</tr>
</thead>
<tbody>
<tr>
<td>@47291271</td>
<td>null</td>
</tr>
<tr>
<td>@57820183</td>
<td>null</td>
</tr>
</tbody>
</table>

Clean-up
Properties
Condition to generate a valid token within a given timeframe

\[ Pr(\text{token generation fails}) < \frac{1}{2^\lambda} \]

Under token uniformity assumption.

- \( n \) number of tokens in circulation
- \( n_{\text{max}} \) total token space size
- \( T \) #trials per timeframe (100 ms)
- \( \lambda \) security parameter
Condition to generate a valid token within a given timeframe

\[ \left( \frac{n}{n_{\text{max}}} \right)^T < \frac{1}{2^\lambda} \]

Under token uniformity assumption.

- \( n \) number of tokens in circulation
- \( n_{\text{max}} \) total token space size
- \( T \) #trials per timeframe (100 ms)
- \( \lambda \) security parameter
Condition to generate a valid token within a given timeframe

\[ n < 2^{\log_2(n_{\text{max}}) - \frac{\lambda}{T}} \]

- \( n \) number of tokens in circulation
- \( n_{\text{max}} \) total token space size
- \( T \) trials per timeframe (100 ms)
- \( \lambda \) security parameter

Under token uniformity assumption.
Condition to generate a valid token within a given timeframe

\[ n = 2^{\log_2(n_{max}) - \frac{\lambda}{T}} \]
Condition to generate a valid token within a given timeframe

\[ n = 2^{\log_2(n_{max}) - \frac{\lambda}{T}} \]
Condition to generate a valid token within a given timeframe

\[ n = 2^{\log_2(n_{\text{max}}) - \frac{\lambda}{T}} \]
Proof of uniformity of the token generation

HMAC SHA3-224 is uniform

HMAC SHA3-224 mod $10^8$ remains uniform.
Proof of uniformity of the token generation

HMAC SHA3-224 is **uniform**

Total: $2^{224}$ evenly distributed

Remainder $\approx 2^{24}$

Quotient $\approx 2^{200}$

modulo $10^8 \approx 2^{26.5}$. 


Proof of uniformity of the token generation

HMAC SHA3-224 is uniform

\[ R \approx 2^{24} \]

\[ Q \approx 2^{200} \]

\[ \text{modulo } 10^8 \approx 2^{26.5} \]

\[ \Pr(\text{pathological}) = \Pr(Q \cdot 10^8 \leq x < 2^{224}) \]

\[ = \frac{R}{2^{224}} \]

\[ = \frac{10,294,216}{2^{224}} \approx \frac{2^{23.3}}{2^{224}} \approx \frac{1}{2^{200}} \]
## Performances

### Expected memory usage

<table>
<thead>
<tr>
<th>Field</th>
<th>Memory Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Card number + CVC</td>
<td>64 bits</td>
</tr>
<tr>
<td>Expiry date</td>
<td>64 bits</td>
</tr>
<tr>
<td># of use(s)</td>
<td>8 bits</td>
</tr>
<tr>
<td>Rand</td>
<td>32 bits</td>
</tr>
</tbody>
</table>

Total memory usage: \(= 168 \text{ bits}\)
Performances

Expected memory usage

Card number + CVC: 64 bits
Expiry date: 64 bits
# of use(s): 8 bits
Rand: 32 bits

Total: 168 bits

256 bits [AES]
### Performances

<table>
<thead>
<tr>
<th>Theoretical performances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fill rate before possible failure with probability over $2^{-512}$ and $T=400,000$</td>
</tr>
<tr>
<td>Maximum RAM used without encryption</td>
</tr>
<tr>
<td>Maximum RAM used with encryption</td>
</tr>
</tbody>
</table>

10M rows x 168 bits
## Performances

<table>
<thead>
<tr>
<th>Theoretical performances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fill rate before possible failure with probability over $2^{-512}$ and $T=400,000$</td>
</tr>
<tr>
<td>Maximum RAM used without encryption</td>
</tr>
<tr>
<td>Maximum RAM used with encryption</td>
</tr>
</tbody>
</table>

- Maximum RAM used: 10M rows x 256 bits
## Performances

<table>
<thead>
<tr>
<th>Real performances</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Fill rate before failure</td>
<td>99.998%</td>
</tr>
<tr>
<td>Maximum RAM used without encryption</td>
<td>3.2 GB</td>
</tr>
<tr>
<td>Maximum RAM used with encryption</td>
<td>25.6 GB</td>
</tr>
</tbody>
</table>
# Performances

<table>
<thead>
<tr>
<th></th>
<th>Real performances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fill rate before failure</td>
<td>99.998%</td>
</tr>
<tr>
<td>Maximum RAM used without encryption</td>
<td>3.2 GB</td>
</tr>
<tr>
<td>Maximum RAM used with encryption</td>
<td>25.6 GB</td>
</tr>
</tbody>
</table>

Average tokenization: 0.041ms

#trials / timeframe: 7221

```c
struct row {
    unsigned long long cb;
    int r;
    unsigned char useNumber;
    unsigned long long t;
    short extra1;
    int extra2;
};
```
Extra possibilities
Data sustainability:
1. Logging every transaction
2. Data cleaning: check expiry time & nb. of max. uses
Unforgeability

1. Assessing the customer’s identity
2. Customer’s anonymity guaranteed
Multi-factor authentication

Client claim their identity from their device.

Unforgeability
Hash tables
And space efficient data-structures

- Hash tables instead of basic tables to handle duplicate values
- Space-efficient data structures: Bloom and Cuckoo filters
Hash tables

And space efficient data-structures

- Hash tables instead of basic tables to handle duplicate values
- Space-efficient data structures: Bloom and Cuckoo filters
Demo
Takeaway slide

• **Collisions ?**
  7000 tries per seconds to be optimized

• **Uniformity ?**
  Proven

• **Untraceability ?**
  No link by design

• Extra possibilities: auditable DB, unforgeability, scalable Bloom filters
Thank you for your attention.

Questions?

Shoutout to Pascal and Olivier

paul.huynh@loria.fr
simon.masson@loria.fr
cyrius.nugier@laas.fr
diane.leblanc-albarel@irisa.fr
agathe.blaise@lip6.fr
brice.wandji@fime.fr