PAYMENT PROJECT

Principal Investigator: Pro. Steven Murdoch

Research Fellow: Aydin Abadi

UCL
PAYMENT PROJECT

Progress so far

PAYMENT

Categories

- Authorised Push Payment (APP) Fraud
- Privacy Enhancing Technology
- Cryptocurrencies

Results

- Designed a Protocol for Payment with Dispute Resolution
- Developed Attacks on Private Set Intersection
- Designed a protocol for Fair Payment System for Cloud’s Storage

Paper

www.rephrain.ac.uk
"Authorised Push Payment" (APP) fraud:

- Definition: An APP fraud is a type of cyber-crime where a fraudster tricks a victim into making an authorised online payment into an account controlled by the fraudster.
- It is called “authorised” because the victim authorises the payment.
- The APP fraud has various variants, such as:
  - romance
  - investment
  - CEO
  - invoice
• The amount of money lost due to APP frauds is substantial

  • Only in the first half of 2021, a total of £355 million was lost to APP frauds.

• APP fraud is a global issue.

  • According to the FBI’s report, victims of APP frauds reported at least a total of $419 million losses, in 2020.

  • Recently, Interpol warned its member countries about a concerning variant of APP fraud called investment fraud via dating software.
Although the UK’s regulators (unlike other countries) have provided specific guidelines to financial institutes to prevent APP frauds occurrence and improve victims’ protection, these guidelines are:

- ambiguous
- open to interpretation

Also, there exists no mechanism in place via which honest victims can prove their innocence.
To protect victims of APP frauds, we proposed:

1. **Formal Definition**: we put forward the notion of “Payment with Dispute Resolution” (PwDR):
   - Identified a PwDR scheme’s core security properties:
     1. security against a malicious victim.
     2. security against a malicious bank.
     3. privacy.
   - formally defined the PwDR scheme.

\[
\begin{align*}
\text{keyGen}(1^\lambda) &\rightarrow (sk, pk) \\
\text{bankInit}(1^\lambda) &\rightarrow (T, pp, l) \\
A(1^\lambda, T, pp, l) &\rightarrow \hat{m}_1^{(c)} \\
\text{insertNewPayee}(\hat{m}_1^{(c)}, l) &\rightarrow \hat{l} \\
\text{genWarning}(T, \hat{l}, aux) &\rightarrow \hat{m}_1^{(b)} \\
A(T, \hat{l}, \hat{m}_1^{(b)}) &\rightarrow \hat{m}_2^{(c)} \\
\text{makePayment}(T, \hat{m}_2^{(c)}) &\rightarrow (\hat{z}, \hat{\pi}) \\
\forall j, j \in [m] : \\
\left( \text{verComplaint}(\hat{z}, \hat{\pi}, \hat{m}, \hat{l}, j, sk_D, aux, pp) \rightarrow \hat{w}_j \right) \\
\text{resDispute}(T_2, \hat{w}, pp) &\rightarrow v = [v_1, ..., v_4]
\end{align*}
\]
2. **Efficient Protocol**: we designed an efficient protocol that realises the PwDR’s definition.

   - formally proved the protocol is secure (i.e., meets the formal definition).

3. **Analysed the Protocol’s Cost**: we performed a cost analysis of the construction via both asymptotic and runtime evaluation (via a prototype implementation).
PAYMENT PROJECT
Protecting Victims of APP Frauds
Main Tools We Used

• The PwDR Protocol's building blocks:
  • Commitment scheme
  • Digital signature
  • Smart contract
  • Pseudorandom function
  • Bloom filter
  • Threshold voting protocols
PAYMENT PROJECT
Protecting Victims of APP Frauds
The PwDR Protocol’s Workflow

Payment Phase

1. Key agreement
2. Confirmation
3. New payee’s detail
4. Read
5. Warning/pas
6. Read
7. Payment Request
8. Read
9. Payment Confirmation
PAYMENT PROJECT
Protecting Victims of APP Frauds
The PwDR Protocol’s Workflow

Dispute Resolution

1. **Complaint**
   - Customer

2. **Read**
   - Arbiter
   - Arbiter
   - Arbiter
   - Arbiter

3. **Vote**
   - Arbiter
   - Arbiter
   - Arbiter
   - Arbiter

4. **Read**
   - Dispute Resolver

5. **Don’t Reimburse**
   - Reimburse

Smart Contract: Verifiable protocol

**PAYMENT PROJECT**
Protecting Victims of APP Frauds
The PwDR Protocol’s Workflow
# PAYMENT PROJECT
## Protecting Victims of APP Frauds

### The PwDR Protocol's Complexity

<table>
<thead>
<tr>
<th>Party</th>
<th>Setting</th>
<th>Computation Cost</th>
<th>Communication Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$e = 1$</td>
<td>$e &gt; 1$</td>
<td></td>
</tr>
<tr>
<td>Customer</td>
<td>✓</td>
<td>✓</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>Bank</td>
<td>✓</td>
<td>✓</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>Arbiter $D_1, \ldots, D_{n-1}$</td>
<td>✓</td>
<td>✓</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>Arbiter $D_n$</td>
<td>✓</td>
<td>✓</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>Dispute resolver</td>
<td>✓</td>
<td>✓</td>
<td>$O(n)$</td>
</tr>
</tbody>
</table>

$n$: number of arbiters

$e$: threshold
## PAYMENT PROJECT

**Protecting Victims of APP Frauds**

The PwDR Protocol’s Runtime in Millisecond

### Table 1:
The PwDR’s runtime (in ms). Broken-down by parties. In the table, \( n \) is the number of arbiters and \( e \) is the threshold.

<table>
<thead>
<tr>
<th>Party</th>
<th>( n = 6 )</th>
<th>( n = 8 )</th>
<th>( n = 10 )</th>
<th>( n = 12 )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( e = 1 )</td>
<td>( e = 4 )</td>
<td>( e = 1 )</td>
<td>( e = 5 )</td>
</tr>
<tr>
<td>Arbiter ( D_n )</td>
<td>0.019</td>
<td>0.220</td>
<td>0.033</td>
<td>0.661</td>
</tr>
<tr>
<td>Dispute resolver ( DR )</td>
<td>0.001</td>
<td>0.015</td>
<td>0.001</td>
<td>0.016</td>
</tr>
</tbody>
</table>

\( n \): number of arbiters  
\( e \): threshold
• To protect victims of APP frauds, we proposed “Payment with Dispute Resolution” (PwDR) scheme.

• We hope that our result lays the foundation for future solutions that will protect victims of APP frauds.
The end