TimeTrust

RISE Second Annual Conference
London, 21st of November 2019

Ioana Boureanu

Tom Chothia
University of Birmingham

Liqun Chen
University of Surrey

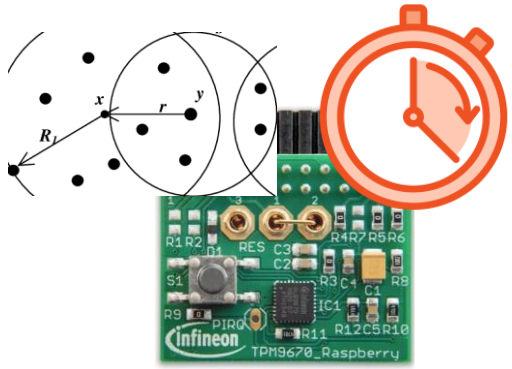
TimeTrust

Robust Timing via Hardware Roots of Trust and Non-standard Hardware
-with Application to EMV Contactless Payments-

Topic: -- using **hardware-roots of trust** to build (~backwards-compatible) counteractions of attacks based on **forging proximity** and/or **forging time checks**

-- focus on **contactless payments** made with bankcards and smart devices (phones)

-- focus on a **formal treatment**, with some practical assessment



Academic Parties

Ioana Boureanu (Lecturer in Secure Systems)

Liqun Chen (Professor in Secure Systems)

Tom Chothia (Senior Lecturer in Cyber Security)

Industry partners/advisors: Main representatives

Mike Ward (Director, Product Development and Innovation, EMV & Digital Devices)

Charles Yuexi Chen (Senior Director - EMV Solutions)

Fraser Dickin (Senior Research Engineer)

Steve Pannifer (Director)

TimeTrust-funded Researchers

Postdoctoral Researcher (Univ. of Birmingham) – Andreea-Ina Radu
-- to start Jan. 2020 until June 2021
-- 18 months

Postdoctoral Research (Univ. of Surrey) -- Dr. Chris Newton
-- to start in March 2020, until October 2021
-- 20 months

TimeTrust Objectives

Objective 1 (via WP2) **--- Security against collusive-relaying** (**CR)**, by using and extending HW-RoT such as **TPMs**

Objective2 (via WP3) -- **Security against CR** for non-standard hardware (ns-HW) such as mobile devices, via **vTPMs and TEEs** and hardware surfaces

Objective 3 (via WP4) **– Privacy in proximity checking**, using HW-RoT and/or vTPMS/ TEEs and/or ns-HW

Objective 4 (via WP5) -- Cryptographic and formal-verification analyses

Objective 5 (via WP6) -- application to **contactless EMV**

- to inform future work into enhancing the security of

EMV against untrustworthy readers

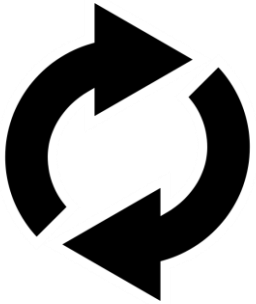
One of TimeTrust's Scopes
-- wireless communication & proximity checking

UNIVERSITY OF SURREY

-- relay attacks 
-- proximity checking = the main way to hinder relay
-- present in one EMV contactless payment protocol (Mastercard's RRP)

Would it collude in relaying to take illicit payments ?

!! ALSO, no track in EMV that a reader would do the proximity-checking



time0 ———  ———  ———→

←——— time1

RTT:= time1- time0

? RTT <= bound ?

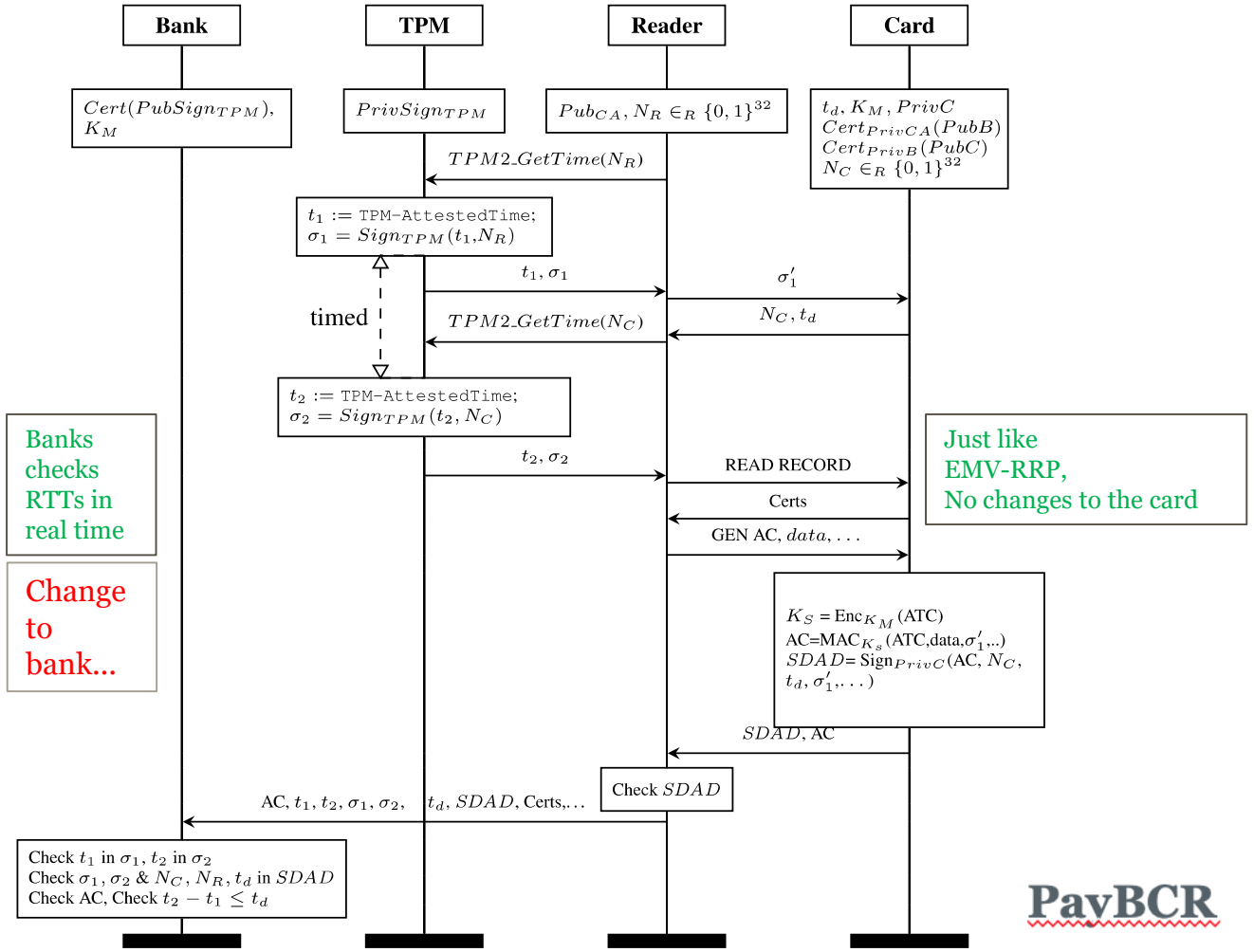
New security notion: **collusive relaying**

FinancialCrypto 19 *[CBC]*

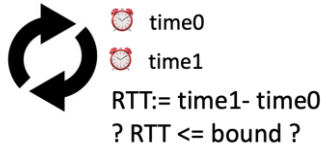
-- introducing PayBCR, PayCCR: ~backwards-compatible RRP + resistance to collusive relaying

Basically, a TPM-based proximity checking mechanism , via two GETTIME() commands.

We do not think we can do efficiently in 1 command, due to TPM being stateless



Our Threat Model for These

- On... distance-bounding 

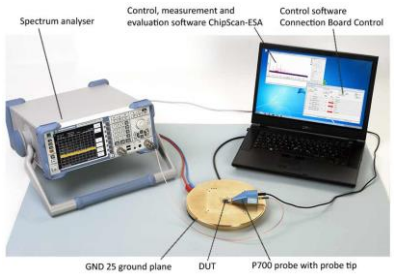
+ EMV  + Trusted computing 



- Attacker can
  1. Corrupt cards, incl. all their key material, use these at any location
  2. Know the readers' key/secret material, control their software, **within the communication model**
  3. **Not** corrupt the TPM in ways unspecified by TCG:
     a. cannot tamper with the initial setup of the TPM's Clock;
     b. cannot mount any physical attack on the TPM's time-reporting TPM-AttestedTime=(Clock, Time)
     c. can deviate the TPM's Clock **only by making it go forward** w.r.t. to the real time **by a negligible fraction.**
- Std. assumptions: cryptographic primitives are secure as per their threat model

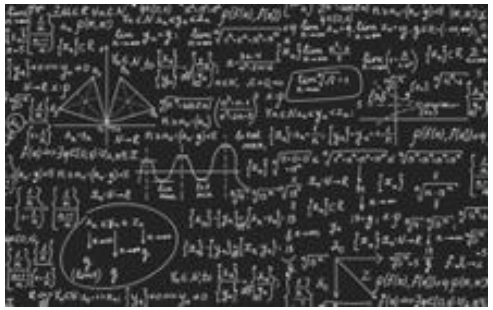
Progress

1. Practice



- Implemented a robust **RRP-relaying app**
- Implemented PayBCR with a TPM 2.0 v1.38, in collaboration with Consult Hyperion
- Tested it against relaying, collusive-relaying and comparing it with RRP

2. Formal



- A first, prototype modelling of RRP in the cryptographic prover (no full TPM specs yet ...) – **EasyCrypt**
- Developed a new symbolic formalism to formally verify PayBCR-like protocols + automatic analysis in **ProVerif**
- ...

Symbolic Verification in Proverif & Cryptographic Proofs in EasyCrypt

```
    new chal:channel;
    out(cP, chal);
    out(SecChal(chal), (tpmID, xnR, sigma1));
    (***)
    out(cP,(time, sigma1));
    0.

et Card(cID:bitstring,tpmID:bitstring) =
    in(cP, CertC:bitstring);
    let Test1 = equals(pubKey(cID), Verify(CertC,pubKey(idBank))) in
    in(cP, CertTPM:bitstring);
    let Test2 = equals(pubKey(tpmID), Verify(CertTPM,pubKey(idBank))) in

    in(cP, xsigma1:bitstring);
    new nC:bitstring;
    event Action(cID, xsigma1, nC);
    out(cP, nC);

    in(cP, (xt2:bitstring, xsigma2:bitstring, xt1:bitstring, xnR:bitstring, xcertTPM:bitstring));
    let Test3 = equals(xcertTPM, CertTPM) in

    out(cP, CertC);
    in(cP, xGen:bitstring);
    let Test4 = equals(xGen, GEN_AC) in
    let (temp:bitstring,a:bitstring) = Verify(xsigma1, pubKey(tpmID)) in
    let (temp:bitstring,b:bitstring) = Verify(xsigma2, pubKey(tpmID)) in
    let TestSigma1 = equals(xnR ,a) in
    let TestSigma2 = equals(nC ,b) in

    event EndC(cID, tpmID, xsigma1, xsigma2);

    let Ks = sEnc(ATC, shk(cID, idBank)) in
    let AC = Mac((ATC, xsigma1, xsigma2), Ks) in
    let SDAD = Sign((AC, xsigma1, xsigma2), secKey(cID)) in
    out(cP, (SDAD, AC));
    0.
```

```
module type Card ={
  proc init(): (pkey*skey)
  proc generateNO(): message
  proc certify(sk: skey,un: message): (message * signature)
  proc send(un: message,sk: skey): (message * signature)
}.

module type Oracle ={

  proc adv_send_as_card (t: ctime, sid:session_id, m: message): unit
  proc hon_card_send (t: ctime, sid: session_id): unit
  proc adv_send_as_reader (t: ctime, sid: session_id, m: message): unit
  proc hon_reader_send (t: ctime, sid: session_id): unit
  proc card_read (t: ctime, sid: session_id): message
  proc reader_read (t: ctime, sid: session_id): message
  proc adv_read(t: ctime, sid: session_id): message
  proc verify_read(sid: session_id): bool option
```

-- main difficulty modelling Euclidean spaces and timing
-- also, modelling the TPM….

Future

- Improve the PayBCR implementation, leveraging at different TPM implementations (... discussions with *Infineon* re GETTIME())

- Look at Visa's L1 solution for relaying

- Look at practical timing-attacks on TPMs/TEEs
- Use TPMs for EMV-aspects other than relays...

- ....

THANK YOU!

For further questions, contact me at
i.boureanu@surrey.ac.uk