Private Authentication with Alpha-Beta Privacy

Laouen Fernet Sebastian Mödersheim

DTU Compute
{lpkf,samo}@dtu.dk

June 15, 2023

Increasing digitalization

Health

Payment

Transport

Voting

- Strong guarantees of privacy
- More subtle than secrecy

Alpha-Beta Privacy

- Declarative and intuitive, based on logic
- Payload α
- Technical information β
- Tool support

$\alpha \equiv age \geq 18$ $\quad$ $\alpha \equiv v_1, \ldots, v_n \in \{0,1\} \wedge \Sigma_{i=1}^{n} v_i = 42$

Unlinkability: $x_1, \ldots, x_n \in \text{Tags}$

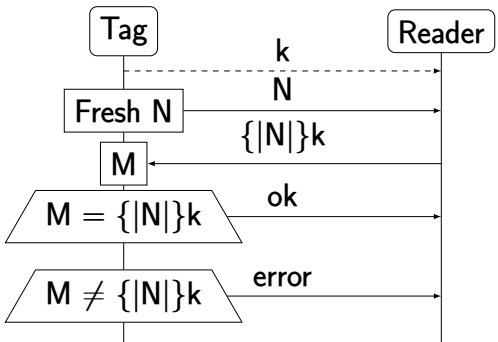
This Paper

- New specifications for BAC and Private Authentication
- Discussion of modeling challenges
- Explanation of the tool output

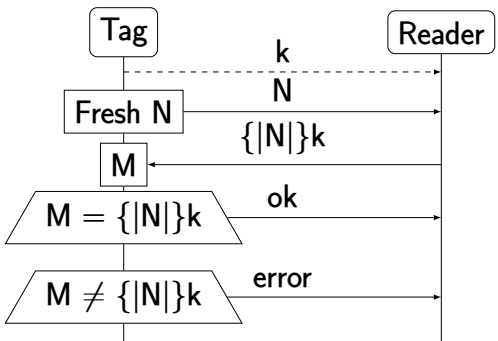
Passport Reader

CBP International Travel Preclearance Operations in Canada, U.S. Customs and Border Protection, public domain

BAC: Challenge

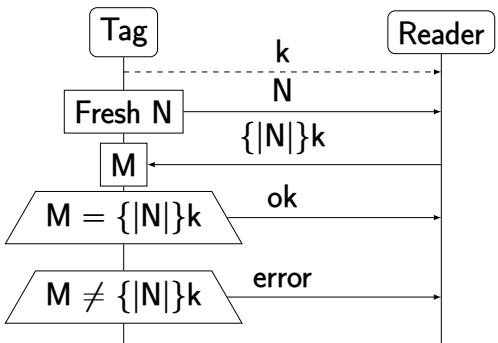


BAC: Challenge



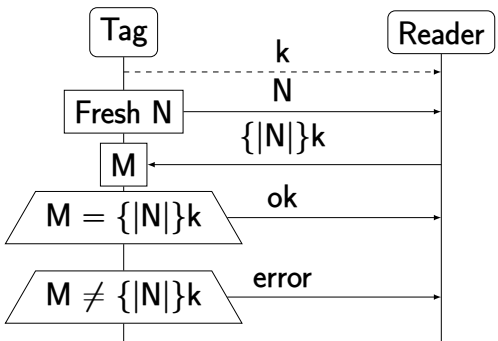

```
Transaction Challenge:
* Tag in {t1,t2}.
new Nonce.
send session(Tag,Nonce).
send Nonce.
send scrypt(sk(Tag),Nonce)
```

BAC: Response



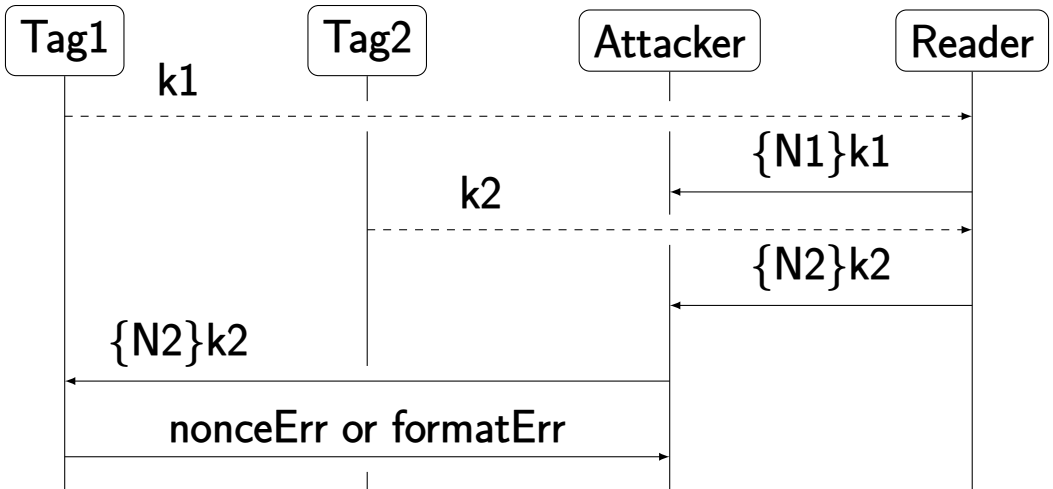

```
Transaction Response:
receive Session.
try Tag   = sfst(Session) in
try Nonce = ssnd(Session) in
receive M.
try
 N = dscrypt(sk(Tag),M)
in
...
```

BAC: Response




```
...
State := noncestate[Nonce].
if Nonce = N and
   State = fresh then
  noncestate[N] := spent.
  send ok
else  send nonceErr
catch send formatErr
```

BAC: Attack Trace Tag1 $\stackrel{?}{=}$ Tag2



BAC: Attack Trace $\quad \mathsf{Tag1} \stackrel{?}{=} \mathsf{Tag2}$

2 Challenge transactions: $\alpha \equiv \mathsf{Tag1}, \mathsf{Tag2} \in \{\mathsf{t1}, \mathsf{t2}\}$

$\mathsf{I1} \mapsto \mathsf{session}(\mathsf{Tag1}, \mathsf{Nonce1}) \quad \mathsf{I2} \mapsto \mathsf{scrypt}(\mathsf{sk}(\mathsf{Tag1}), \mathsf{Nonce1})$

$\mathsf{I3} \mapsto \mathsf{session}(\mathsf{Tag2}, \mathsf{Nonce2}) \quad \mathsf{I4} \mapsto \mathsf{scrypt}(\mathsf{sk}(\mathsf{Tag2}), \mathsf{Nonce2})$

BAC: Attack Trace $\quad \mathsf{Tag1} \stackrel{?}{=} \mathsf{Tag2}$

2 Challenge transactions: $\alpha \equiv \mathsf{Tag1}, \mathsf{Tag2} \in \{\mathsf{t1}, \mathsf{t2}\}$

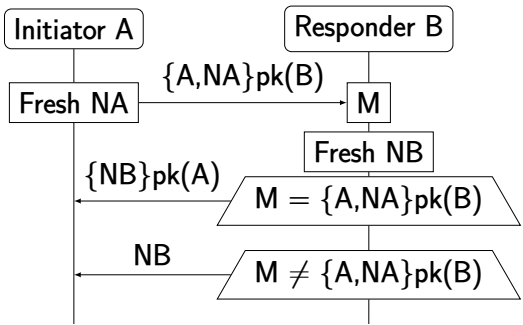
$\mathsf{I1} \mapsto \mathsf{session}(\mathsf{Tag1}, \mathsf{Nonce1}) \quad \mathsf{I2} \mapsto \mathsf{scrypt}(\mathsf{sk}(\mathsf{Tag1}), \mathsf{Nonce1})$

$\mathsf{I3} \mapsto \mathsf{session}(\mathsf{Tag2}, \mathsf{Nonce2}) \quad \mathsf{I4} \mapsto \mathsf{scrypt}(\mathsf{sk}(\mathsf{Tag2}), \mathsf{Nonce2})$

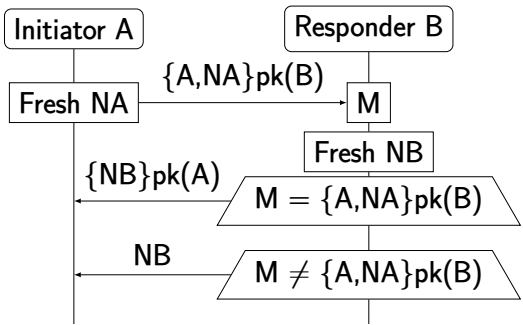
The intruder uses session I1 but encryption I4:

- decryption works iff $\mathsf{Tag1} = \mathsf{Tag2}$
- nonceErr if $\mathsf{Tag1} = \mathsf{Tag2}$
- formatErr if $\mathsf{Tag1} \neq \mathsf{Tag2}$

AF0: Initiator

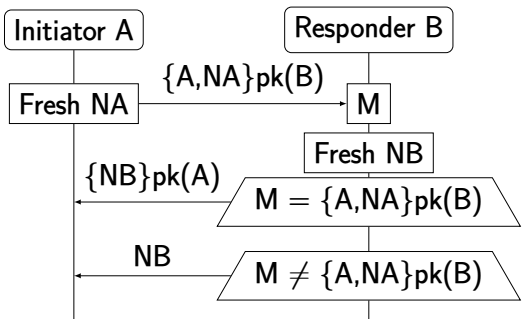


AF0: Initiator




```
Transaction Initiator:
* A in {a,b}.
* B in {a,b,i}.
if B = i then new NA,R.
  send crypt(pk(B),
             pair(A,NA),R).
  * A = gamma(A) and
    B = gamma(B)
else new NA,R.
  send crypt(pk(B),
             pair(A,NA),R).
  * B in {a,b}
```

AF0: Responder




```
Transaction Responder:
* B in {a,b}.
receive M.
try
 DEC = dcrypt(inv(pk(B)),M)
   A = proj1(DEC) in
if A = i then new NB,R.
  send crypt(pk(A),NB,R).
  * B = gamma(B)
else new NB,R.
  send crypt(pk(A),NB,R)
catch new NB. send NB
```

AF0: Attack Trace

In the paper :)

Conclusions

- (α, β)-privacy for real protocols: declarative, intuitive
- Rediscovery of attacks and better understanding of privacy guarantees
- Useful to find strongest goal