Abhi Devireddy
CS 4821 – Computer Security

Security Presentation Outline

- What is authentication?
- Current forms of authentication:
    - 1 factor – Passwords
    - 2 factor –Biometric or Dynamic PIN
    - 4 factor authentication – Someone you know
- Hashing functions
    - Geo-hashing
    - Time hash
    - SecurID hashing function
- Cryptanalysis of the RSA SecurID
- Threats to the security of the SecurID
- Usefulness of the SecurID token

The presentation will be about the cryptographic security of the RSA SecurID token. The token is a small electronic device with an operating speed of less than 1MHz. The token has a LCD panel by means of which it can display a pseudorandom sequence of numbers which change at regular intervals.

The algorithm used to generate these pseudorandom set of numbers is proprietary and was reverse engineered by cryptanalysts in 2002. The reverse engineered algorithms were verified to be the same as RSA's implementation. There also exist many software implementations which can produce the token codes for different platforms.

We will be talking about the hash algorithm used in the generation of the pseudorandom set of numbers. The SecurID token uses a 64bit time value to generate a new code every 30 or 60 seconds.

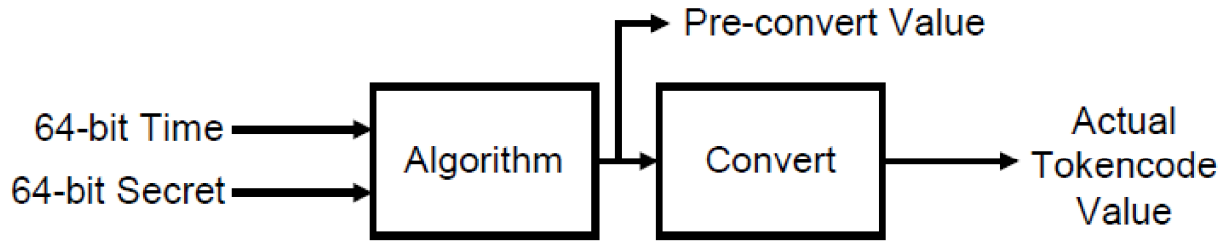
Given the simplicity of the hardware used in the SecurID tokens, and the usage of a unique 64-bit identifier for every user, the set of outputs from the token seem to be limited. Even more so when we notice that the 64-bit time retains only 24-bit form the current time value and replicates some bits.

Given enough past token codes, the attacker could then generate a token code at any future instance. Thus the security of the system could be compromised.

SecurID Algorithm for 64-bit time value

```
INT64 time64;
INT32 time;
UCHAR byte;
time = gettimeofday(); // Seconds since 01/01/86, 00:00
// Round down time
time = time / 30;
time = time / 4;
time = time * 4;
// Expand time into 64-bits, duplicate least significant byte
byte = time | 0xFF;
time = time << 8;
time = time | byte;
time64 = time;
time64 = time64 << 32;
time64 = time64 | time;
```

Thus we notice the loss of entropy in the time value used in the computation of the output. The other 64-bit input depends on the token that is currently being used and is used as an identifier.



After this, ewe would be talking about the possible attacks on the SecurID token and the exploits that have been made public in recent years.

We will also discuss the usefulness of the token and how the avalanche effect and other properties of encryption algorithms can make MITM attacks very difficult.